

ISSN 0234-0453

ИНФОРМАТИКА И ОБРАЗОВАНИЕ

1988





Во время приемочных испытаний

Сегодня в номере — КУВТ УКНЦ

Опытный образец принят Государственной комиссией

Интервью: новый комплекс с трех точек зрения

Двухпроцессорная ЭВМ на рабочем месте учащегося

Бейсик, Фортран, Паскаль, Си, Рапира, Е-практикум...

**Школьная компьютеризация требует изменения
организации производства**

**10 000 часов наработки на отказ —
цель ближайшего будущего**



ИНФОРМАТИКА И ОБРАЗОВАНИЕ

Содержание

КУВТ УКНЦ: опытный образец принят 3

Методика обучения

Шень А. Информатика в IX классе 8

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПРОВЕДЕНИЮ ПРАКТИ-
КУМА В X КЛАССЕ 15

Зайдельман Я. О двух последних темах X класса 30

Сочнев С. Изучение алгоритмического языка в курсе информатики 38

Коновалов Л. Из опыта преподавания темы «Табличные величины.
Алгоритмы работы с табличными величинами» 42

Кобринский Я., Штильман Б. Программа факультативного курса
«Избранные главы информатики» 44

Сергеева Т., Чернявская А. Дидактические требования к компьютерным
обучающим программам 48

КВТ

Юнерман Н. Использование графической системы ШПАГА при введении
основных понятий программирования 52

Козловский С. Узловые проблемы оценки качества изображения на
экране дисплея 55

Глазова Н., Голубев С., Магазов С., Сергеева Т. Структура и функции
обучающей системы 59

Бобко И., Плотников А., Сапрыкина Г. КУВТ-86 в подготовке работников
народного образования 61

Стандарт на ПЭВМ 64

Педагогический опыт

Городилова Г., Самородническая П., Аллахвердиев Р. Компьютеры и
орфография 67

Шальнов А. Проблемы компьютерного обучения в вузе 75

Жебрак М. Использование ЕС ЭВМ в обучении 77

Диминштейн О., Думитрашку С., Полануер М. Программируемые микро-
калькуляторы в педагогическом институте 79

К итогам конкурса

Белешапка В. Конкурс на учебник информатики	98
Гегечкори А. Ретроспективные замечания о конкурсе учебников	107

Внеклассная работа

Дуванов А., Зайдельман Я., Первин Ю. Роботландия	109
Боканс Я., Грибустс А. Не только на уроке информатики	117
Лилитко Е. О первом всесоюзном турнире по компьютерной игре «Бой в памяти»	120

Точка зрения

Маслов А. О содержании курса информатики	123
Петров М. Нужен ли в спецшколе «Агат»?	125

Репортаж номера

Новый центр информатики	126
-------------------------	-----

Веселый урок

128

2

Обложка Э. Бажиллина

В оформлении номера принимали участие Э. Бажилин, Р. Бумагин, А. Лобашинский, А. Пономарев, С. Расторгуев.

Главный редактор академик
В. А. МЕЛЬНИКОВ
 Редакционная коллегия
И. М. БОБКО
Б. М. ГЕРАСИМОВ
Г. В. ГОДЖЕЛЛО
А. В. ДЕНИСЕНКО
А. П. ЕРШОВ
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

Редактор отдела *Ф. Даниловский*
 Редактор отдела *К. Шеховцев*
 Научный редактор *Т. Драгньш*
 Заведующая редакцией *Н. Игнатова*
 Художественный редактор *Л. Розанова*
 Корректор *Н. Минервина*

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

Почтовый адрес: 107847, Москва, Лефортовский пер., 8
 Телефон редакции: 249-97-77

Сдано в набор 27.11.87. Подписано в печать 29.12.87. А 07477. Формат 70×100/16. Печать офсетная. Усл. печ. л. 10,40. Уч.-изд. л. 12,85. Усл. кр.-отт. 42,88. Тираж 93 660 экз. Заказ 3339. Цена 60 коп.

Ордена Трудового Красного Знамени Чеховский полиграфический комбинат ВО «Союзполиграфпром» Государственного комитета СССР по делам издательств, полиграфии и книжной торговли. 142300, г. Чехов Московской области.

КУВТ УКНЦ: опытный образец принят

В октябре 1987 г. завершила работу Государственная комиссия по приемке опытного образца нового школьного комплекса учебной вычислительной техники УКНЦ («Электроника МС 0202»). Сейчас, когда вы читаете эти строки, первые экземпляры нового КУВТ уже установлены в школах, разворачивается его массовый выпуск. Предлагаем вашему вниманию описание комплекса и несколько интервью с участниками приемочных испытаний.

В состав КУВТ УКНЦ входят рабочее место преподавателя (РМП), 12 рабочих мест учащихся (РМУ), локальная информационная сеть со скоростью передачи данных 57 кбод и блок электропитания с выходным напряжением 42 В.

Как в РМП, так и в РМУ входят 16-разрядные ЭВМ «Электроника МС 0511» и мониторы; в РМП, кроме того, есть накопитель на гибких магнитных дисках НГМД 6022 и принтер типа D-100. К любой ЭВМ можно подключить кассетный магнитофон.

«Электроника МС 0511» включает 2 однокристалльных процессора K1801BM2 производительностью более 600 тыс. оп./с каждый. Первый процессор осуществляет основную обработку информации, второй реализует символьно-графические функции, обслуживает периферийные устройства. Оба они программно доступны.

Вот еще несколько характеризующих



МС 0511 цифр: емкость ОЗУ центрального процессора 56К байт, периферийного процессора — 32К байт, видеопамять — 96К байт, встроенное ПЗУ на 32К байт, возможность подключения внешнего ПЗУ на 48К байт; разрешающая способность экрана — 640×288 точек, на нем помещается 26 строк текста по 32, 64 или 80 символов в каждой.

Структурная и программная совместимость МС 0511 с рядом СМ ЭВМ, ДВК, мини-ЭВМ «Электроника 100/25», «Электроника-79» позволит использо-

вать в ней программные продукты, созданные для этих ЭВМ.

При конструировании МС 0511 и подготовке ее производства были использованы последние достижения отечественной техники и технологии: комплект заказанных БИС на 3 тысячи вентилях в кристалле, СБИС ОЗУ на 64К бит, однокристалльный микропроцессор. На подготовившемся к производству этой ЭВМ заводе действуют автоматизированные производства, включающие механическую обработку деталей и печатных плат на станках с ЧПУ, поточную линию химической обработки печатных плат, автоматическую линию пайки, отмывки и формовки выводов, программно управляемое оборудование контроля качества — все это не только повышает производительность, но и позволяет добиться высокого качества продукции.

В базовое программное обеспечение КУВТ УКНЦ входят операционная система РАФОС, управляющая программа локальной сети, языки программирования Бейсик, Фортран, Паскаль, Модула-2, Си, ассемблер, макроассемблер, готовятся Рапира, Е-практикум. Имеются базовые пакеты прикладных программ, включающие редакторы текстов, программы подготовки документов, информационно-справочные системы, «электронную таблицу», пакеты прикладных программ для специализированных применений.

Сегодня на страницах нашего журнала еще раз встретятся люди, вместе работавшие во время приемочных испытаний: один из создателей «Электроники МС 0202» А. А. Попов, председатель Государственной комиссии академик А. П. Ершов и член комиссии главный инспектор Управления информатики и электронно-вычислительной техники Минпроса СССР О. Ф. Титов.

— Александр Анатольевич, чем для вас, разработчика, важен новый комплекс школьных ЭВМ?

— Ну, о технических данных говорить не будем, они у вас есть, об особенностях производства вы тоже знаете... А вот известно ли вам, что

КУВТ УКНЦ для нас — завершение первого ряда конструкций, предназначенных для обучения компьютерной грамоте? Начинается этот ряд с «Лунохода» — игрушки для 4—6-летних детей: яркая, ездит, пульт управления... Так вот в ней — 4-разрядный микропроцессор! И с ее помощью можно изучать все основные элементы программирования. Я говорю вполне серьезно. Ведь что ждет выпускника школы? Один из возможных вариантов (который, кстати, становится все более частым) — работа оператором автоматизированного рабочего места, где ему придется манипулировать функциональной клавиатурой, общаясь со своим АРМ на его языке. И пусть «Луноход» не станок с ЧПУ, но числовое программное управление у него есть, и пульт функциональной клавиатуры тоже.

Другой возможный вариант — выпускник будет сталкиваться с вычислительной техникой в качестве пользователя (это может случиться и с бухгалтером, и с журналистом, и с начальником цеха, и с музыкантом). Тогда он будет работать на ПЭВМ, используя меню возможных операций, написанное на его «профессиональном» языке. А подготовиться к этому можно на сравнительно простых машинах, таких как БК-0010. Я думаю, школьные ЭВМ уже пора продвигать в средние и даже начальные классы — на БК-0010 (это основа КУВТ-86) в наших подшефных школах четвероклассники работают и с пользой, и с удовольствием. Да и дома этот компьютер может быть очень полезен.

Кстати, БК-0010 ругают часто — мол, плохой компьютер, и памяти мало, и вообще... А давайте вспомним, с чего ЕС ЭВМ начинались! Одна из первых машин этого ряда ЕС 1020 имела ОЗУ на 64К байта, и работали на ней не любители, а профессионалы, и хорошо работали. А ведь у БК-0010 — 32К, всего в два раза меньше!

И вот теперь последняя наша разработка — УКНЦ. Этот комплекс — первый, который мы специально разрабатывали для школ, с его помощью старшекласники смогут сделать очень

многое. И если кто-то из них выберет третий вариант жизненного пути — станет программистом, то с большой вероятностью ему придется работать на машинах типа СМ или «Электроника», а они совместимы с УКНЦ, так что школьные уроки будут очень кстати.

— *ЭВМ — сердце комплекса, но без периферийного оборудования она мало на что способна. Кстати, и к БК-0010 претензии в первую очередь из-за отсутствия у него дисководов.*

— Вопрос о периферийном оборудовании очень непрост. Его изготавливает множество предприятий разных ведомств. Возьмем дисководы: без дискет они будут простаивать, а качество отечественных дискет такое, что работать с ними очень трудно. Так что периферия — это проблема комплексная, межотраслевая, и решать ее надо комплексно.

— *А что вы скажете о качестве мониторов? Некоторые медики опасаются, что они могут испортить зрение школьников.*

— Мониторы делятся на два класса. Для первого нужны специальные электронно-лучевые трубки, прецизионное изготовление; они дают изображение, сравнимое по качеству с фотографическим. Такие мониторы используются в особых случаях — например, в автоматизированном рабочем месте технического редактора, АРМ конструктора-дизайнера и т. п. Они выпускаются малыми партиями. Второй класс — обычные терминалы. У нас, например, за такими работают операторы САПР. Вначале, когда САПР только установили, бывали различные негативные эффекты — переутомление, нарушение работы вестибулярного аппарата. Мы тогда ввели перерывы для обязательной гимнастики для глаз* — и с тех пор все в порядке.

Надо еще заметить, что качество монитора определяется по самому мелкому шрифту, который он способен

воспроизвести. Но всегда ли надо таким пользоваться? На «Ямахе», например, буквы высотой 10—15 мм. Тут уже все равно — бытовой телевизор подключен к ЭВМ или специальный монитор. Вот с расчетом на это и надо делать обучающие программы для школ, а не давать подростку по 80 символов в строке, переписывая на экран учебник.

— *Какими вы видите перспективы школьных ЭВМ?*

— Предела совершенству, конечно, нет. Однако до какой степени разумно улучшать параметры компьютеров? Уже 32-разрядные ЭВМ позволяют программисту работать, не думая о том, как устроена машина. Видимо, школьная ЭВМ будущего должна быть 32-разрядной, и думаю, что задача создания такого компьютера будет поставлена перед нами если не в конце этой пятилетки, то в начале следующей.

— *Андрей Петрович, расскажите, пожалуйста, нашим читателям о работе приемочной комиссии.*

— Прежде всего следует отметить, что работа комиссии не сводилась просто к анализу конкретной конструкции; она отражала также и общие процессы, происходящие в стране, в частности, изменение взаимоотношений между производителем и потребителем. Нельзя сказать, что гегемония производителя (в нашем случае разработчика) уже в прошлом, однако ситуация заметно улучшилась. Мы убедились в том, что у потребителя есть возможность оказывать необходимое влияние на характер взаимоотношений с разработчиком, на процедуру приемочных испытаний и т. п.

Вспомним, что этот КУВТ предъявлялся, по существу, дважды (первый раз в конце 1986 г., тогда по нему было сделано много замечаний), но это печальное на первый взгляд обстоятельство оказалось серьезным стимулом как для разработчиков, так и для заказчиков. После первых испытаний удалось наладить реальное сотрудничество со-

*Комплекс упражнений для глаз опубликован в № 5 нашего журнала за 1987 г.— *Прим. ред.*

здателей программного обеспечения (прежде всего из МГУ — группа Кушниренко) и разработчиков ПЭВМ; интенсивно поработав, разработчикам удалось ко второму предъявлению устранить большую часть недостатков.

Конечно, работу над данной конструкцией можно было бы продолжать еще долго. Однако здесь важно было соблюсти верные пропорции — каждый месяц задержки выпуска КУВТ УКНЦ означал бы производство вместо них КУВТ-86, которые, хотя и могут уже сейчас производиться в большом количестве, не решают проблем школьной компьютеризации, поддержки курса информатики, не отвечают требованиям сегодняшнего дня и тем более перспективным. Так что до бесконечности затягивать доводку УКНЦ тоже нельзя.

Часть вопросов должна быть решена после начала серийного производства. Собственно, комиссия принимала не ЭВМ как изделие, а скорее конструкцию ЭВМ. Предъявленные экземпляры произвели неплохое впечатление, меры по организации производства — тоже, и было решено рекомендовать серийный выпуск КУВТ УКНЦ. Конечно, это некий аванс, и было установлено, что для первой партии машин (в начале 1988 г.) будет организована опытная эксплуатация в реальной обстановке интенсивного учебного процесса по программе, согласованной с Минпросом СССР и Госпрофобром. Это позволит сразу устранить недостатки, которые будут замечены, и, главное, обеспечить необходимую надежность. Ведь ненадежная ЭВМ — худший из всех возможных вариантов.

Пока вопрос надежности исследован не полностью. Правда, был организован первоначально не планировавшийся «прогон» (непрерывная работа КУВТ в течение суток), который прошел в целом неплохо. Это дало комиссии уверенность, что от предъявленной конструкции можно будет потребовать тех высоких показателей надежности, которые были записаны в техническом задании — 10000 часов наработки на отказ для отдельной ПЭВМ и порядка 5000 часов — для КУВТ в целом. Было, правда, и особое мнение, но в ито-

ге мы решили, что удостоверение надежности может быть проведено в период опытной эксплуатации.

У происходящего есть еще один, очень важный аспект. Ведь школьный компьютер — это пробный камень смены технологии производства. Тираж КУВТ УКНЦ таков, что его можно выпускать только на автоматизированных предприятиях, создание которых — дело даже более важное, чем разработка конкретной модели ПЭВМ.

Подытожить можно так: комиссией проделана серьезная, я бы сказал, вьедливая работа по знакомству с новым КУВТ. Так что в целом я с оптимизмом смотрю на перспективу появления в 1988 г. этого комплекса в массовом производстве.

— Можно ли надеяться, Андрей Петрович, что на ближайшие годы проблемы школьной компьютеризации будут с помощью КУВТ УКНЦ решены?

— Нет. Проблемы еще останутся. УКНЦ не решит всех задач профессиональной подготовки, кроме того, объем выпуска этого КУВТ будет пока недостаточен. Но там, где он будет доступен, он станет технической основой для решения проблем компьютеризации. Ведь компьютеризация — не только ЭВМ, но и периферийные устройства (в частности, хорошие мониторы, учебные роботы, которых пока нет, разного рода датчики, модемы для связи между удаленными ЭВМ), а главное — программное обеспечение. КУВТ УКНЦ — это «минимальная» техническая база, достаточная для разработки на ней всего комплекса программных средств, предусматриваемого программой компьютеризации образования на ближайшие годы.

— Олег Федорович, вы записали в приемочном акте особое мнение о ненадежной надежности КУВТ УКНЦ. Что было причиной этого?

— Появление у нас первых образцов персональных ЭВМ, разрабатывавшихся специально для средних учебных заведений всех типов — очень важное событие. Оно приближает тот момент,

когда в большинстве школ компьютер как объект изучения в курсе ОИВТ дополнится компьютером — новым мощным инструментом изучения школьного курса. Вместе с тем ждут нас и новые заботы.

В настоящее время серьезные нарекания вызывает явно недостаточная надежность поступающей в школы вычислительной техники. Помимо прямого ущерба качеству учебного процесса это приводит к чрезмерным эксплуатационным расходам. В связи с тем, что возможности модернизации уже установленного в школе оборудования весьма ограничены, особенно важно своевременно принять меры по повышению безотказности и ремонтпригодности комплексов учебной вычислительной техники.

Необходимо отметить, что уже сегодня компьютеризация образования является одним из стимулирующих факторов развития отраслей промышленности, производящих ВТ, периферийное оборудование, машинные носители информации, средства связи. Возникла необходимость формирования инфраструктуры, обеспечивающей широкомасштабное производство надежных и безопасных учебных компьютеров, электромонтажные работы по их установке во вновь создаваемых кабинетах школ, техническое обслуживание и ремонт. Сжатые сроки создания такой инфраструктуры, большая ее стоимость, недостаточный опыт применения ЭВМ в школе и крайний дефицит обоснованных рекомендаций по повышению надежности персональных ЭВМ в условиях реальной эксплуатации в общеобразовательной школе предъявляют повышенные требования к «запасу прочности», заложенному на этапе раз-

работки. Возможности министерства-разработчика в этом плане вполне достаточны, и необходимо использовать их в полной мере, тем более что речь идет о компьютере, которому предстоит, возможно, стать одним из самых массовых в текущей пятилетке.

Действующий с 01.07.87 г. стандарт (ГОСТ 27201-87 «Машины вычислительные электронные персональные») определяет среднюю наработку на отказ для базового комплекта персональной ЭВМ, соответствующего рабочим местам учащихся КУВТ (включая клавиатуру, монитор и средства подключения к локальной сети), не менее 10 тыс. часов. Это практически годы безотказной эксплуатации. Разумеется, достижение такой надежности для серийно выпускаемой школьной ЭВМ является 7 новой задачей, ее решение потребует и в дальнейшем совместных систематических усилий изготовителя и разработчика при необходимом содействии головной организации по персональным ЭВМ и соответствующих НИИ АПН СССР.

КУВТ УКНЦ (КВУ «Электроника МС 0202») в целом по функциональным возможностям соответствует требованиям технического задания. Вместе с тем представленные комиссии опытные образцы (все компоненты которых могли быть многократно проверены и испытаны разработчиком — это не массовая продукция) имеют, по моей оценке, значительно более низкую наработку на отказ, чем требует стандарт. Предстоит еще немало поработать, чтобы устанавливаемые в школе компьютеры оправдали надежды учителей и учащихся.

Материалы подготовил К. ШЕХОВЦЕВ

А. ШЕНЬ

Информатика в IX классе

8 Построение алгоритмов для решения задач из курса математики. В УП разбираются три математические задачи: вычисление значения многочлена по схеме Горнера, приближенное вычисление площадей и поиск корня.

По поводу первой задачи мы ограничимся единственным комментарием к приведенному в УП алгоритму. Его работу легче понять, если иметь в виду, что после любого числа циклов выполнения команды повторения (с. 63) выполняется такое соотношение:

$$y = (\dots((a[0]x + a[1])x + a[2])x + \dots + a[i-1])x + a[i]$$

(т. е. сделано i шагов вычислений по схеме Горнера, вплоть до коэффициента $a[i]$). В начале $i=0$ и $y=a[0]$. На каждом шаге мы увеличиваем i на 1 и выполняем еще один этап вычисления по схеме Горнера. Цикл завершается, когда $i=n$, т. е. использованы все коэффициенты многочлена вплоть до $a[n]$.

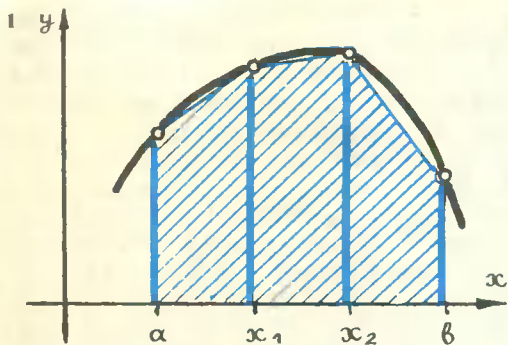
Второй пример — традиционная задача численного интегрирования, т. е. приближенного нахождения интеграла. Однако слово «интеграл» в УП заменяется словами «площадь криволинейной трапеции». Рассматриваемый метод вычислений называется «метод прямоугольников». На практике он применяется редко, так как дает меньшую точ-

ность, чем другие (ничуть не более сложные) методы. Кроме того, у школьников может вызвать недоумение то обстоятельство, что он не дает точного ответа даже в случае, когда криволинейная трапеция является обычной «прямолинейной» трапецией (т. е. для функции $f(x)=ax+b$).

Поэтому можно предложить разобрать другой алгоритм приближенного вычисления площадей — метод трапеций. Его идея в том, что мы заменяем кривую ломаной, составленной из отрезков прямых (рис. 1). При этом криволинейная трапеция заменяется фигурой, состоящей из нескольких обыкновенных трапеций, площадь которой будет равна сумме площадей трапеций. Подсчитаем эту сумму, предположив, что интересующий нас отрезок $[a, b]$ разбит на n равных частей точками x_1, x_2, \dots, x_{n-1} . В этом случае высота h каждой трапеции равна $(b-a)/n$, основания равны $f(x_i)$ и $f(x_{i+1})$ (а площадь трапеции равна произведению высоты на полусумму оснований). Например, для $n=3$ (изображенный на рис. 1 случай) получаем

$$S = \frac{h}{2}f(a) + hf(x_1) + hf(x_2) + \frac{h}{2}f(b).$$

Видно, что все значения функции входят в эту формулу с одинаковыми коэффициентами, кроме двух крайних, коэффициенты при которых вдвое меньше (это не удивительно: ведь значе-



ния функции в промежуточных точках влияют на площадь и слева, и справа от себя, а значения функции в крайних точках — только в одну сторону). Получаем такой алгоритм (вначале вычисляются члены, соответствующие крайним точкам, а затем добавляются слагаемые для внутренних точек):

алг ПЛОЩАДЬ (вещ a, b, s , цел n)

арг a, b, n

рез s

нач вещ h, x , цел i

(h — высота трапеций, i — число учтенных внутренних точек,

x — первая еще не учтенная внутренняя точка; s — сумма по крайним и первым i внутренним точкам)

$h := (b - a) / n$

$i := 0$

$x := a + h$

$s := (f(a) + f(b)) * h / 2$

пока $i \neq n - 1$

нц

$s := s + f(x) * h$

$x := x + h$

$i := i + 1$

кц

кон

Видно, что этот алгоритм имеет практически ту же сложность, что и алгоритм метода прямоугольников, приведенный в УП. Точность его, однако, гораздо выше (грубо говоря, он дает вдвое больше верных цифр при том же числе n отрезков деления). Существуют и другие, еще более точные алгоритмы — метод парабол (формула Симпсона), интегрирование по нулям ортогональных многочленов (метод Гаусса) и др. Разработка и анализ таких алгоритмов

являются задачами раздела математики, называемого вычислительной математикой.

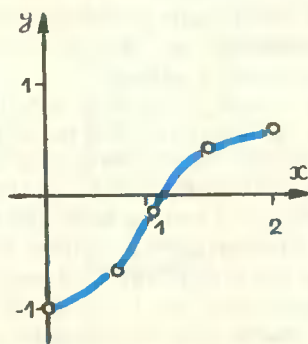
К этому же разделу математики при-
мыкает следующий пример из УП —
отыскание приближенных значений кор-
ней. Эта задача делится на две части
— «отделение» корня (поиск интер-
вала, в котором он заключен) и «уточ-
нение» (нахождение значения корня с
нужной точностью). Решение первой
части требует общих сведений о свой-
ствах уравнения, корень которого ищется;
ничего содержательного в рамках
школьного курса о ее решении ска-
зать нельзя. Напротив, вторая часть
может стать предметом подробного об-
суждения на уроке.

В курсе математики постоянно ре-
шают самые разные уравнения, т. е. **9**
ищут их корни (если они, конечно,
есть). На самом деле эти уравнения
специально подобраны так, чтобы их
можно было решить теми методами, ко-
торые проходят на уроках математики.
Стоит чуть-чуть отклониться от стан-
дартных типов уравнений, и мы сразу
окажемся беспомощными. Попробуйте,
например, найти корень у $x \sin x - 1 = 0$.

Может быть, решение не удастся най-
ти потому, что его нет? Давайте
составим таблицу значений левой части
этого уравнения ($x \sin x - 1$) при несколь-
ких значениях x :

x	0,0	0,5	1,0	1,5	2,0
$x \sin x - 1$	-1,0	-0,75	-0,15	0,49	0,81

Нарисуем примерный график функции
 $y = x \sin x - 1$ (рис. 2), отметив на рисунке
точки из приведенной таблицы (пунктир-



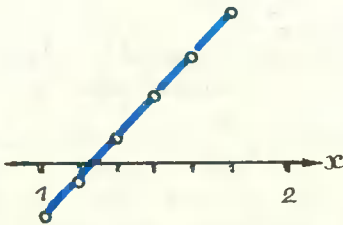
ная линия проведена наугад через эти точки). Корень уравнения $x \sin x - 1 = 0$ — это такое x , при котором $x \sin x - 1$ обращается в 0, т. е. график пересекает ось абсцисс. Посмотрев на график, мы видим, что участок от 1,0 до 1,5 является «подозрительным», поскольку при переходе от 1,0 к 1,5 выражение $x \sin x - 1$ меняет знак: из отрицательного становится положительным. Изучим этот участок графика более подробно.

x	1,0	1,1	1,2	1,3	1,4	1,5
$x \sin x - 1$	-0,15	-0,02	0,12	0,25	0,38	0,49

Похоже, что корень находится где-то между 1,1 и 1,2 (рис. 3).

3

10



На самом деле существует математическая теорема, из которой следует, что действительно на участке от 1,1 до 1,2 имеется корень нашего уравнения. Эта теорема утверждает, что если функция f такова, что $f(a) \leq 0$, $f(b) \geq 0$, причем f непрерывна (это означает, грубо говоря, что график f — линия, не имеющая разрывов), то на отрезке $[a, b]$ имеется такое x , что $f(x) = 0$. В нашем случае $f(x) = x \sin x - 1$, $a = 1,1$, $b = 1,2$. Функция $f(x) = x \sin x - 1$, как доказывается в курсах высшей математики, непрерывна. Поэтому на отрезке $[1,1, 1,2]$ наше уравнение имеет корень (на самом деле этот корень равен 1,114157...).

Итак, что же получается? Уравнение не решается методами, разбираемыми в курсе математики, но имеет корень. Как же этот корень найти?

Вернемся на минуту к обыкновенным квадратным уравнениям. Пусть, например, нам нужно изготовить стержень длиной x сантиметров, где x — корень уравнения $x^2 - x - 1 = 0$ (у него два корня, но один отрицателен и потому отпадает). Решая это уравнение по известной формуле, получаем: $x = (1 + \sqrt{5})/2$. Но эта форма ответа может удовлетворить

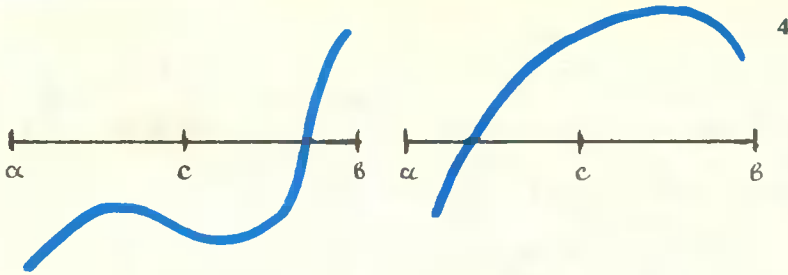
нас, лишь пока мы занимаемся математикой. Как только мы начинаем изготавливать стержень длиной x сантиметров, мы убеждаемся, что на линейке нет деления « $(1 + \sqrt{5})/2$ см», а есть лишь сантиметровые и миллиметровые деления. Что же делать? Надо посмотреть в таблице — или вычислить с помощью микрокалькулятора — приближенное значение $\sqrt{5}$. Оно равно 2,236068...; подставляя его, получаем, что $x = 1,6180...$ Теперь остается лишь отмерить линейкой нужный отрезок стержня. (При этом наша точность даже избыточна: вряд ли на линейке есть более мелкие, чем миллиметровые деления, так что хватило бы двух цифр после запятой.)

Мы искали $\sqrt{5}$ в таблице или пользовались калькулятором. Но как узнали составители таблиц, что $\sqrt{5}$ примерно равен 2,23606...? Для калькулятора вопрос звучит так: что делает программа, управляющая калькулятором, когда мы нажимаем клавиши, требующие вычислить $\sqrt{5}$?

Изложим один из возможных способов приближенного нахождения $\sqrt{5}$. Будем искать его «подбором». Нам надо найти такое x , что $x^2 = 5$. Попробуем $x = 2$. Тогда $x^2 = 4$. Это мало. Пробуем $x = 3$. Тогда $x^2 = 9$. Это много. Значит, нужное x расположено где-то посередине между 2 и 3. Пробуем $x = 2,5$. Тогда $x^2 = 6,25$. Много. Значит, нужное x где-то между 2 и 2,5. Пробуем $x = 2,25$, тогда $x^2 = 4,415625$. Мало. Значит, x между 2,125 и 2,25. Пробуем середину: $x = 2,1875$, $x^2 = 4,785...$ — снова мало, значит, x между 2,1875 и 2,25, пробуем $x = 2,21875$, $x^2 = 4,922...$, снова мало, x между 2,21875 и 2,25, пробуем середину: $x = 2,234375$, $x^2 = 4,9824...$, снова мало, x между 2,234375 и 2,25.

Этот процесс можно продолжать бесконечно, но остановимся здесь. Что мы получили? Мы узнали, что $\sqrt{5}$ лежит между 2,23 и 2,25, так что если мы примем его равным 2,24, то совершим ошибку не более одной сотой. Если такая точность нас устраивает, то на достигнутом можно остановиться. Если же нет, то можно продолжать описанный процесс поиска корня делением пополам, пока не достигнем нужной точности.

Подобный способ отыскания корня



можно применить и к уравнению $x \sin x - 1 = 0$, а также к многим другим уравнениям вида $f(x) = 0$, где $f(x)$ — какое-то выражение, содержащее x . Назовем отрезок $[a, b]$ отрезком перемены знака, если $(f(a) \leq 0 \text{ и } f(b) \geq 0)$ или $(f(a) \geq 0 \text{ и } f(b) \leq 0)$. Эквивалентное определение: $[a, b]$ — отрезок перемены знака, если $f(a) \cdot f(b) \leq 0$.

Основой рассматриваемого способа отыскания корней является такое замечание: если $[a, b]$ — отрезок перемены знака, а c — его внутренняя точка, то один из отрезков $[a, c]$ и $[c, b]$ является отрезком перемены знака. На рис. 4 слева отрезком перемены знака является $[c, b]$, справа $[a, c]$.

Способ отыскания корня состоит в том, что мы делим отрезок перемены знака пополам и выбираем из двух его половин ту, которая сама является отрезком перемены знака. Этот процесс повторяется до тех пор, пока длина отрезка не станет достаточно малой. После этого середина полученного отрезка считается приближенным значением корня.

Алгоритм этот приведен в библиотеке алгоритмов (А11). При его исполнении после любого цикла отрезок $[a, b]$ является отрезком перемены знака. Команда ветвления действует так: если левая половина отрезка является отрезком перемены знака, то мы сужаем отрезок, сдвигая влево его правый конец (рис. 5). В противном случае отрезком перемены знака является правая половина исход-

ного отрезка и мы сужаем отрезок, сдвигая его левый конец (рис. 6).

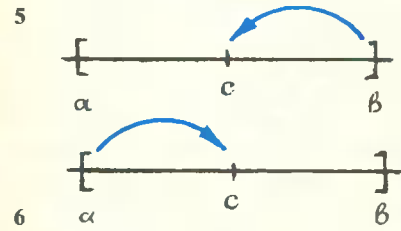
После завершения исполнения команды повторения входящее в нее условие не соблюдается, т. е. $b - a \leq 2\epsilon$, и середина отрезка $[a, b]$ (т. е. число $(a+b)/2$) является приближением к корню с точностью не хуже ϵ .

Для каких функций имеет смысл изменять наш алгоритм? Мы уже говорили, что функция f должна менять знак. В этом случае в ходе исполнения алгоритма будет найден отрезок изменения знака, имеющий малую длину, и в качестве результата работы алгоритма будет выдана его середина. Правда, для того чтобы мы могли заключить, что найдено приближение к корню, мы должны еще знать, что функция непрерывна. Например, для функции

$$f(x) = \begin{cases} 1, & \text{если } x \in [0, 1] \\ -1, & \text{если } x \in [1, 2] \end{cases}$$

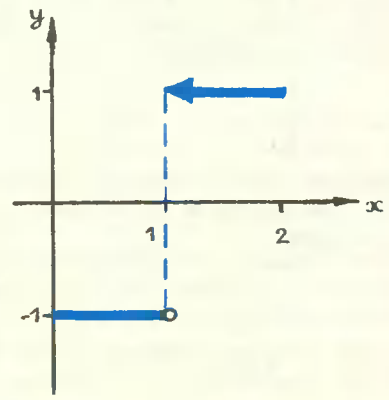
(рис. 7) отрезок $[2, 2]$ является отрезком перемены знака, но корней там нет (данная функция не является непрерывной); применение нашего алгоритма приведет к отрезку малой длины, окружающему точку $x=1$.

Кроме того, уравнение $f(x) = 0$ (в том числе и для непрерывной f) может иметь



5

6



7



несколько корней на отрезке перемены знака (рис. 8). В этом случае наш способ позволит отыскать один из них. Если функция f монотонна (как написано в описании рассматриваемого алгоритма на с. 95 УП), то корень может быть только один. Так что для монотонной непрерывной функции, меняющей знак на некотором отрезке, этот алгоритм действительно позволяет найти ее единственный корень с заданной точностью.

Построение алгоритмов для решения задач из курса физики

12 Этот раздел УП содержит три примера. Примеры трудные — в первую очередь тем, что для их понимания требуется свободное владение соответствующим материалом из курса физики. Если вы почувствуете, что хорошо разобрать их все за отведенное для прохождения этой темы время не удастся, мы рекомендуем ограничиться разбором любых двух (по вашему выбору).

Метод наименьших квадратов. Пример 1 демонстрирует математические методы обработки полученной в результате эксперимента информации на простейшем материале. Приведенные в УП числовые данные могут быть использованы на уроке, если у школьников есть возможность пользоваться непрограммируемыми микрокалькуляторами. Если же такой возможности нет, то нужно выбрать более простой для расчета вариант, иначе вычисления отнимут слишком много времени. Вот один из возможных вариантов.

$$\begin{aligned} U_1 &= 1\text{В} & I_1 &= 1\text{А} \\ U_2 &= 3\text{В} & I_2 &= 2\text{А} \\ U_3 &= 5\text{В} & I_3 &= 4\text{А} \end{aligned}$$

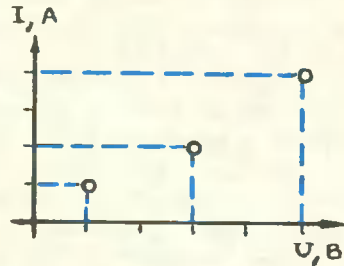
Обсуждая эти данные, следует добиться от школьников понимания следующих вещей.

А. Результаты измерений не соответствуют закону Ома: если вычислять сопротивление по формуле $R=U/I$, то для каждого из трех измерений оно будет свое (1 Ом, 1,5 Ом и 1,25 Ом).

Б. Это несоответствие можно объяснить двумя причинами: либо наши измерения не совсем точные, либо исследуемый «резистор» на самом деле является не резистором, и ток через него зависит от приложенного напряжения не линейно, как того требует закон Ома, а более сложным образом.

В. Выбор между этими двумя объяснениями зависит от точности применяемых нами приборов: чем точнее приборы, тем менее вероятно, что отклонения результатов измерений от теоретической линейной зависимости — всего лишь ошибки.

Г. Если мы пришли к выводу о том, что отклонения от закона Ома объясняются погрешностями измерения, связанными с недостаточной точностью приборов, то для оценки сопротивления резистора следует провести прямую $I=kU$ как можно ближе к отмеченным на графике (рис. 9) точкам и считать сопротивлением резистора число $1/k$.



После того как школьники все это поняли, можно сообщить им формулу для вычисления коэффициента k . Хотя ученикам предлагается готовая формула для вычисления k , происхождение которой остается для них неизвестным, учителю следует знать ее смысл. Если

$$(U_1, I_1), (U_2, I_2), \dots, (U_n, I_n)$$

экспериментальные точки, то с помощью этой формулы можно найти такое k , что

$$(I_1 - kU_1)^2 + \dots + (I_n - kU_n)^2$$

(сумма квадратов отклонений) будет минимальной (это объясняет название «метод наименьших квадратов»). Убедиться в том, что это действительно так, можно следующим образом: раскрыв скобки, получаем квадратный трехчлен относительно k :

$$k^2(U_1^2 + \dots + U_n^2) - 2k(U_1I_1 + \dots + U_nI_n) + (I_1^2 + \dots + I_n^2).$$

Квадратный трехчлен $ak^2 + bk + c$ как функция от k при $a > 0$ принимает наименьшее значение в точке $k = -b/2a$. В данном случае $a = U_1^2 + \dots + U_n^2$, $b = -2(U_1I_1 + \dots + U_nI_n)$, откуда получается приведенная в учебнике формула.

Прежде чем применять формулу метода наименьших квадратов к экспериментальным данным, полезно опробовать ее вместе с учениками на простых примерах. Самый простой пример получится, если есть только одна экспериментальная точка. В этом случае получаем $k = UI/U^2 = I/U$ т. е. прямая $I = kU$ проходит как раз через эту точку. Более сложный пример (разобранный в УП) получится, когда точек несколько, но все они в точности лежат на прямой $I = kU$. И в этом случае формула дает ожидаемый результат.

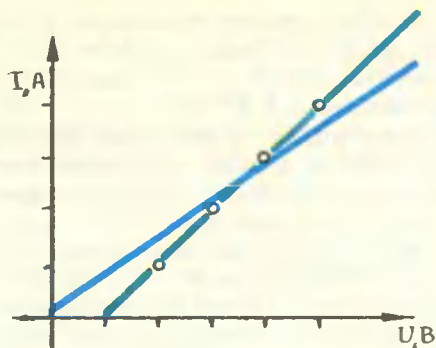
Теперь подставим в формулу имеющиеся экспериментальные данные:

$$k = \frac{U_1I_1 + U_2I_2 + U_3I_3}{U_1^2 + U_2^2 + U_3^2} = \frac{1+6+20}{1+9+25} = \frac{27}{35};$$

$$R = 1/k = 35/27 \approx 1,3 \text{ Ом.}$$

Чтобы убедиться еще раз в правдоподобности полученного ответа, нанесем прямую $I = kU$ на график (рис. 10). Видно, что она действительно проходит близко от экспериментальных точек и что отклонения действительно можно истолковать как погрешности измерения. Можно еще сравнить полученное значение сопротивления (1,3 Ом) с результатами, получающимися из отдельных измерений (1 Ом, 1,5 Ом, 1,25 Ом), и убедиться, что оно действительно лежит где-то посередине между ними.

Главная мысль, которую следует донести до школьников в ходе обсуждения, состоит в том, что при математической обработке результатов экспери-



ментов нельзя ни на минуту забывать о физическом смысле, стоящем за формулами и алгоритмами, иначе можно получить абсурдные результаты. В этом отношении поучителен такой пример.

Пусть в результате измерений мы получили:

$U_1 = 2\text{В}$	$I_1 = 1\text{А}$
$U_2 = 3\text{В}$	$I_2 = 2\text{А}$
$U_3 = 4\text{В}$	$I_3 = 3\text{А}$
$U_4 = 5\text{В}$	$I_4 = 4\text{А}$

13

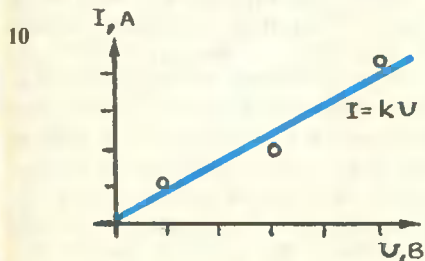
Воспользовавшись формулой метода наименьших квадратов, находим, что $k = 27/20$, $R = 1,35 \text{ Ом}$.

Однако, нанеся соответствующую прямую на график (рис. 11, синяя линия), мы видим, что наблюдается систематическое отклонение экспериментальных точек от этой линии. Зато они точно ложатся на зеленую прямую, задаваемую формулой $I = U - 1$. Хорошо знающие физику ребята сразу сообразят, что такая закономерность может быть объяснена тем, что наш «резистор» на самом деле вовсе не резистор, а батарейка с электродвижущей силой в 1 В и внутренним сопротивлением 1 Ом.

Метод наименьших квадратов применим в тех случаях, когда одна величина изменяется пропорционально другой и нас интересует коэффициент пропорциональности. Среди упражнений к § 8 есть два, которые можно использовать в качестве иллюстрации его применения.

В первом из них длина участка карандаша, покрытого нитью, пропорциональна числу витков. Вот результаты проведенного нами эксперимента (хлопчатобумажные нитки № 40):

Число витков	10	20	30	40	50	60	70	80	90	100
Длина в мм	3	5	8	11	14	16	18	21	24	27



По формуле метода наименьших квадратов получаем, что коэффициент k (в данном случае — толщина нити) равен примерно 0,27 (мм).

Упражнение 3 (измерение толщины листа бумаги) удобно несколько видоизменить, проводя его с книгой и измеряя толщину заданного числа листов в книге с помощью штангенциркуля. Вот результаты такого эксперимента (проведенного с книгой «Изучение основ информатики и вычислительной техники». М.: Просвещение, 1985):

Число листов

5 10 15 20 25 30 35 40 45 50

Толщина, мм

0,4 0,9 1,3 1,8 2,4 2,8 3,3 4,0 4,3 4,9

14 По формуле метода наименьших квадратов находим коэффициент k (толщину листа в миллиметрах), который оказывается примерно равным 0,096.

Колебания шарика на пружине. Этот пример иллюстрирует методы математического моделирования физических процессов. Данный физический процесс описывается дифференциальным уравнением $m\ddot{x} = -kx(t)$ (m — масса шарика, k — жесткость пружины, $x(t)$ — координата шарика в момент t). Оно называется уравнением гармонических колебаний и может быть решено аналитически. При этом получается такой ответ: если в начальный момент времени шарик отклонен на x_0 и отпущен без начальной скорости, то его координата в произвольный момент t находится по формуле

$$x(t) = x_0 \cos(\sqrt{k/m}t).$$

Это точное решение, однако, пока недоступно школьникам. Зато численное, приближенное, решение этого уравнения вполне доступно им уже сейчас. Подобная ситуация складывается и в реальных физических исследованиях: точное аналитическое решение возникающих задач зачастую оказывается невозможным, в то время как приближенное решение с помощью ЭВМ возможно и дает нужный результат с достаточной для нас точностью.

Приступая к разбору примера, полезно сначала предложить ученикам несколько простых вопросов подготовительного характера, например таких.

А. Координата движущегося по прямой шарика в момент t равна x , а скорость равна v . Чему будет равна координата шарика через промежуток времени Δt ? (Ответ: $x + v\Delta t$. Он является приближенным — мы не учитываем изменение скорости шарика; точность приближения тем выше, чем меньше Δt .)

Б. Скорость движущегося по прямой шарика в момент времени t равна v , а ускорение равно a . Чему будет равна скорость шарика через промежуток времени Δt ? (Ответ: $v + a\Delta t$. Он, как и предыдущий, является приближенным: мы не учитываем изменения ускорения шарика.)

Далее полезно напомнить о законе Гука, согласно которому сила, с которой растянутая пружина действует на прикрепленное к ней тело, пропорциональна ее удлинению: $F = -kx$ (знак минус поставлен, так как сила направлена противоположно удлинению); коэффициент k называется жесткостью пружины. После этого в беседе с классом следует добиться понимания следующих трех вещей:

А. В положении равновесия на висящий на пружине шарик действуют две равные по величине и противоположные по направлению силы: сила тяжести и сила упругости. Их сумма (результатирующая сила, действующая на шарик) равна 0, потому-то он и находится в равновесии.

Б. Если отклонить шарик на расстояние x от положения равновесия, то сила тяжести останется неизменной, а к силе упругости прибавится число $-kx$. Таким образом, результирующая сила будет равна $-kx$.

В. Ускорение шарика можно найти, используя второй закон Ньютона: $F = ma$, откуда $a = F/m = -(k/m)x$.

Теперь школьники готовы к заполнению таблицы. Заполнение производится по строкам, строки заполняются слева направо. Ускорение в момент начала интервала определяется по формуле $a = -kx/m$, скорость в середине интервала определяется по ускорению в начале интервала и по скорости в начале интервала (первая строка таблицы) или в середине предыдущего интервала (все следующие строки). Затем эта скорость ис-

пользуется для нахождения координаты к началу следующего интервала. Весь этот процесс подробно описан в УП.

Заполнив несколько строк таблицы и сверив их с УП, можно построить график движения шарика (зависимость координаты шарика от времени). Построив график, следует обратить внимание школьников на характер происходящего процесса: сначала скорость шарика мала (координата меняется медленно, наклон касательной к графику мал). Затем шарик разгоняется (модуль скорости увеличивается, кривая идет все круче); между $t=0,07$ и $t=0,08$ шарик проходит положение равновесия (координата равна нулю, модуль скорости максимален) и начинает замедляться. Эта картина колебаний хорошо знакома школьникам: ее можно наблюдать, например, с помощью качающегося маятника.

Полезно сравнить данные в таблице с полученными по точной формуле $x(t) = x_0 \cos(\sqrt{k/m} t)$.

Номер	1	2	3	4	5	6	7	8	9	10
Точный ответ	1,00	0,98	0,92	0,83	0,70	0,54	0,36	0,17	-0,08	-0,23
Результат вычислений	1,00	0,98	0,92	0,82	0,69	0,53	0,35	0,16	-0,004	-0,24

Видно, что они довольно близки. Это показывает, что выбранное нами приближение (мы считали, что на интервале длиной $\Delta t=0,01$ скорость и ускорение шарика меняются мало) достаточно точно. Даже если мы увеличим интервал времени Δt , взяв его равным 0,02 с, как предлагается в упражнении 5, то все равно согласие между точным решением и приближенным остается достаточно хорошим.

Номер	1	2	3	4	5
Точный ответ	1,00	0,92	0,70	0,36	-0,08
Результат вычислений	1,00	0,92	0,69	0,35	-0,04

Следует подчеркнуть, что излагаемый в данном примере материал весьма труден. По существу, школьники познакомились с простейшим методом численного решения дифференциальных уравнений (он называется методом ломаных Эйлера) и с его помощью численно решили дифференциальное уравнение гармонических колебаний. Многие другие задачи школьного курса физики также могут решаться таким методом (например, нахождение орбит планет). Использование компьютеров для решения физических задач может стать увлекательной темой факультативных занятий.

15

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПРОВЕДЕНИЮ ПРАКТИКУМА В X КЛАССЕ

Е. УТЛИНСКИЙ, Д. СМЕКАЛИН, А. ДОДОНОВ

Знакомство

с прикладным программным обеспечением

Описанию прикладного программного обеспечения в пробном учебном пособии (УП) посвящен фактически только один параграф; это объясняется тем, что оно ориентировано прежде всего на «безмашинный» вариант курса. Однако материал этот чрезвычайно важен как с точки зрения иллюстрирования возможностей ЭВМ и их применений, так и с точки зрения освоения школьниками компьютерной грамоты. Поэтому на практических занятиях необходимо

не только организовать работу учащихся с готовыми программами, но и подробно разобрать теоретические вопросы, связанные с их использованием и областями применения.

Прикладное программное обеспечение определяет реальные возможности применения ЭВМ во всех областях народного хозяйства, не связанных непосредственно с разработкой новых программ. Прикладные программы должны облегчать человеку решение с помощью

ЭВМ различных практических задач. До сих пор, когда учащимся нужно было решить с помощью компьютера какую-либо задачу, они разрабатывали алгоритм ее решения и реализовывали его на алгоритмическом языке или языке программирования. При этом ЭВМ рассматривалась как исполнитель, «понимающий» только относительно простые команды алгоритмического языка (языка программирования).

Учащимся известно, что при решении многих сложных задач удобно выделять в них отдельные подзадачи, решение которых осуществляется с помощью вспомогательных алгоритмов (подпрограмм); в результате может оказаться, что основной алгоритм будет состоять только из команд вызова соответствующих вспомогательных алгоритмов. Таким образом происходит расширение системы команд ЭВМ. Каждая команда вызова может рассматриваться как указание исполнителю совершить одно (но уже более сложное) действие.

Можно сказать, что прикладные программы как раз и представляют собой набор подпрограмм, с помощью которых удобно решать задачи определенного типа. Вызов каждой подпрограммы осуществляется с помощью специальной команды, которая, как правило, может задаваться ЭВМ непосредственно человеком в диалоговом режиме с помощью специальных функциональных клавиш.

Набор команд, которые могут быть заданы компьютеру при работе с конкретной прикладной программой, зависит от типа решаемых задач. Обычно количество таких команд невелико, они легко могут быть освоены и непрограммистом.

В данном практикуме учащимся предлагается ознакомиться с наиболее распространенными видами прикладного программного обеспечения. Эти программы используются специалистами самых различных профессий.

Работа с графическим редактором

Цель. Дать понятие о графическом редакторе, области его применения. Привить основные навыки работы с графическим редактором.

Требования к знаниям и умениям. Учащиеся должны знать назначение и области применения графического редактора, уметь пользоваться графическим редактором при решении конкретной задачи.

Методические указания. На знакомство с графическим редактором программой практических занятий выделено 2 учебных часа. Рекомендуется следующая организация практикума:

1-й урок: назначение графического редактора; основные команды редактора.

2-й урок: дополнительные возможности, предоставляемые графическим редактором; решение задач.

1-й урок. Графический редактор предназначен для построения на дисплее рисунков или чертежей с возможностью их редактирования (внесения изменений), сохранения (записи на магнитный диск или ленту), распечатывания на бумаге.

Для создания графика или рисунка в графических редакторах обычно используются следующие команды:

рисовать отрезок;

рисовать прямоугольник;

рисовать окружность (или часть окружности);

отметить последнюю команду;

очистить экран.

При помощи этих команд, исполняемых графическим редактором, можно построить почти любую геометрическую фигуру или рисунок.

Исполнение каждой команды происходит при нажатии пользователем на определенную клавишу.

Прежде чем приступить к практической работе, учащимся необходимо познакомиться с системой команд конкретного редактора.

Если редактор выполнен на профессиональном уровне, то в нем обязательно существует «подсказка» («помощь»), при обращении к которой на экране дисплея появляются названия функциональных клавиш и соответствующие команды. Обычно обращение к «подсказке» возможно в любой момент работы с редактором. Таким образом, нет необходимости заучивать назначение функциональных клавиш. Вызов

«подсказки», как правило, осуществляется нажатием на клавишу «?» или «H» (help — помощь).

Задание 1. При помощи графического редактора на экране дисплея построить треугольник.

Первое построение лучше провести всем классом, вместе.

На экране при работе с графическим редактором размещаются два курсора (вид этих курсоров может быть различным). Курсоры можно перемещать по экрану при помощи клавиш →, ←, ↑, ↓ в любом направлении. Необходимо дать возможность учащимся произвести эту операцию.

Чтобы нарисовать треугольник, необходимо построить три последовательно соединенных отрезка. Построить отрезок можно, если известны две точки (его начало и конец). Можно задать учащимся вопрос: зачем на экране два курсора? (Для задания начала и конца отрезка.)

Для построения первой стороны треугольника необходимо указать курсорами начало и конец отрезка, затем нажать на функциональную клавишу, выполняющую команду «рисовать отрезок». Для определения соответствующей клавиши обратимся к «подсказке». Прочитав нужное обозначение, вернемся к построению рисунка и исполним команду «рисовать отрезок». На экране появится отрезок. Для построения второго отрезка переместим один из курсоров в конец первого отрезка, а второй — в произвольную точку экрана и исполним команду «рисовать отрезок». После этого рекомендуется обратиться к «подсказке» и найти обозначение клавиши, при помощи которой отменяется последняя команда, затем вернуться к рисунку и выполнить команду «отменить последнюю команду». Далее построим второй и третий отрезки так, чтобы на экране появился треугольник.

Задание 2. Построить окружность с центром в одной из вершин треугольника.

Построение можно провести, задавая учащимся следующие вопросы: сколько точек нужно указать редактору для построения окружности? (Две: центр ок-

ружности и радиус.) Как определить клавишу, выполняющую команду «рисуй окружность»? (Обратиться к «подсказке».)

Далее учащимся предлагается выполнить следующие действия: построить окружность; отменить последнюю команду; построить окружность с центром в другой вершине треугольника.

Задание 3. Построить на экране дисплея прямоугольник.

Размер прямоугольника задается одной из его диагоналей, которая, однако, не рисуется. Построение осуществляется с помощью специальной функциональной клавиши. Стороны прямоугольника будут параллельны границам экрана. Следует обратить внимание учащихся на то, что если диагональ прямоугольника задается вертикальной или горизонтальной, то прямоугольник вырождается в отрезок (в саму диагональ).

Задание 4. Построить с помощью графического редактора 5 вложенных друг в друга квадратов так, чтобы последний выродился в точку.

Домашнее задание. Сделать рисунок, который на следующем практическом занятии необходимо построить при помощи графического редактора за 10 мин.

2-й урок. Задание 1. Построить рисунок, сделанный дома.

Если КУВТ оснащен локальной сетью, то можно лучший рисунок переслать по сети на рабочее место преподавателя (РМП) и распечатать на принтере, сохранив затем на диске.

Если в распоряжении учителя имеются другие графические редакторы, учащимся можно предложить самостоятельное знакомство с новым редактором и выполнение заданий 1—3 первого занятия.

На первом занятии не рассматривались такие команды графических редакторов:

- закрасить фигуру;
- изменить цвет рисуемой фигуры;
- изменить цвет фона;
- рисовать закрашенный прямоугольник;
- рисовать символ;
- вывести на экран координаты курсора;

команды работы с внешними устройствами.

Их изучение — на усмотрение учителя. Один из вариантов — демонстрация на РМП.

Рассматривая эти команды, следует учитывать, что в некоторых редакторах они могут отсутствовать.

Их исполнение связано с некоторыми сложностями. Так, например, выполнение команды «изменить цвет рисуемой фигуры» на монохромном дисплее не всегда можно различить. Команда «закрасить фигуру» имеет следующие особенности: цвет линии, ограничивающей закрашиваемую фигуру (бордюрной линии), должен совпадать с цветом закрашки, линия должна быть замкнута. Если одно из этих условий не соблюдается, происходит закрашивание всего экрана.

18

Работа с текстовым редактором

Цель. Дать понятие о текстовом редакторе, области его применения. Дать основные навыки работы с текстовым редактором.

Требования к знаниям и умениям.

Учащиеся должны знать назначение и область применения текстового редактора, уметь использовать текстовый редактор для конкретной задачи.

Методические указания. На знакомство с текстовым редактором выделено 2 учебных часа. Рекомендуется следующая организация практикума:

3-й урок: назначение текстового редактора; основные команды редактора.

4-й урок: дополнительные возможности, предоставляемые текстовым редактором; решение задач.

3-й урок. Текстовый редактор предназначен для набора на экране дисплея различных текстов, схем, таблиц, их редактирования, сохранения на диске, выведения на бумагу.

Начать знакомство с текстовым редактором рекомендуется с команды вызова «подсказки» на экран. Если используемый редактор не дает такой возможности, учащимся следует раздать распечатку команд, выполняемых редактором.

Для редактирования текста обычно

используются следующие команды:

удалить символ в позиции курсора;

удалить символ слева от курсора;

вставить символ;

вставить пустую строку;

удалить строку;

запомнить строку;

воспроизвести запомненную строку.

Некоторые из команд редактирования учащимся известны по практическим занятиям с Е-практикумом и системой программирования.

Задание 1. Отредактировать текст на экране по образцу.

ТЕКСТ

О, смолько нам отккрытий чудных

И гений — парадоксов друг.

Готовит просвещенья дух.

и опыт — сын ошиибибок трудных,

(А. С. Пушкин)

ОБРАЗЕЦ

О, сколько нам открытий чудных

Готовит просвещенья дух.

И опыт — сын ошибок трудных,

И гений — парадоксов друг.

(А. С. Пушкин)

Совместно с учениками необходимо выявить допущенные ошибки в тексте, определить, при помощи каких команд редактора они могут быть исправлены, и произвести редактирование.

В предложенном тексте две ошибки на замену символа (смолько, и), две ошибки на удаление лишних символов (отккрытий, ошиибибок), две ошибки в расположении строк.

Исправление ошибок с заменой символа и удалением символов должно быть уже известно учащимся.

Для изменения порядка строк необходимо: запомнить переносимую строку; воспроизвести запомненную строку в нужном месте; удалить ее со старого места.

Задание 2. Набрать на экране таблицу умножения на 8.

Задание 3. Продублировать на экране таблицу умножения на 8 и превратить ее в таблицу умножения на 9.

4-й урок. **Задание 4.** Оно должно содержать текст, 30—40 строк, с ошибками, подготовленный заранее и разосланный по сети. Если локальной се-

ти в КВТ нет, объем корректируемого текста должен быть уменьшен.

Учащиеся должны самостоятельно отыскать ошибки и отредактировать текст.

Текстовые редакторы могут обладать дополнительными командами редактирования, например:

- задать формат;
 - форматировать абзац;
 - запомнить символ;
 - воспроизвести запомненный символ;
 - задать шаблон;
 - поиск по шаблону;
 - откатиться назад;
 - откатиться вперед;
 - работа с внешними устройствами.
- Поясним их назначение.

Читая различную литературу, мы сталкиваемся с различными форматами (количество символов в строке) написания текста (текстовые колонки газет, текст в энциклопедии, словаре, книге). Перед человеком часто возникает задача перевода текста из одного формата в другой. Она может быть реализована при помощи команд «задать формат» (ей мы задаем редактору количество символов в строке), «форматировать абзац» (при ее выполнении текст абзаца переводится в указанный формат).

Команды «запомнить символ» и «воспроизвести запомненный символ» выполняются аналогично командам «запомнить строку», «воспроизвести запомненную строку» с той лишь разницей, что запоминается и воспроизводится один или несколько символов.

При написании текстов зачастую приходится искать тот или иной символ или слово. Эта операция очень трудоемка и чревата ошибками. В текстовом редакторе для ее выполнения используются команды «задать шаблон» и «поиск по шаблону». Командой «задать шаблон» редактору задается набор символов (слово), который нужно отыскать в тексте. При исполнении команды «поиск по шаблону» редактор находит в тексте заданный набор символов.

Команды «откатиться назад» и «откатиться вперед» позволяют восстановить текст таким, каким он был до коррек-

тировки, и снова восстановить откорректированный текст.

Рассмотрение дополнительных команд зависит от имеющегося программного обеспечения.

З а д а н и е 2. Произвести последовательное форматирование текста (он может быть произвольным) на 50 символов в строке, затем на 30, 40 и 60 символов.

Возьмем за исходный следующий текст.

Текстовый редактор предназначен для набора на экране дисплея различных текстов, схем, таблиц, их редактирования, сохранения текстов на диске, выведения текста на бумагу.

После его форматирования на 30 символов на экране получим следующее:

19

Текстовый редактор предназначен для набора на экране дисплея различных текстов, схем, таблиц, их редактирования, сохранения текстов на диске, выведения текста на бумагу.

Выполнив задание до конца, учащиеся вернут тексту первоначальный вид.

З а д а н и е 3. В предложенном тексте отыскать заданное слово.

Если, например, взять текст из предыдущего задания и задать шаблон «текст», то при выполнении команды «поиск по шаблону» курсор остановится на слове «Текстовый», затем — «текстов», «текстов», «текста».

Работа с учебной базой данных

Цель. Дать понятие об учебной базе данных, области ее применения. Привить основные навыки работы с учебной базой данных.

Требования к знаниям и умениям. Учащиеся должны знать назначение и область применения базы данных, уметь пользоваться учебной базой данных при выполнении поставленной задачи.

Методические указания. На знакомство с учебной базой данных выделено

2 учебных часа. Рекомендуется следующая организация практикума.

5-й урок: назначение учебной базы данных; основные команды учебной базы данных.

6-й урок: дополнительные возможности, предоставляемые учебной базой данных; решение задач.

5-й урок. Учебная база данных предназначена для хранения больших объемов информации и удобного и быстрого доступа к ней.

Основной компонент базы данных — банк данных, в котором и хранятся данные.

Для доступа к информации, хранящейся в банке данных, в базе данных используются различные наборы информационно-поисковых подпрограмм.

Обычно в базе данных используются следующие команды:

- задать формат банка данных;
- создать банк данных;
- задать ключ сортировки;
- сортировать;
- задать ключ поиска;
- поиск.

Поясним приведенные команды базы данных на примерах. Зададим формат базы данных (табл. 1). Задание формата и его редактирование происходят аналогично работе в текстовом редакторе.

Таблица 1

1	2	3	4	5	6
Фамилия	Имя	Физика	Химия	ОИВТ	

Создадим банк данных (табл. 2).

В банк данных можно внести дополнительные данные или произвести редактирование имеющихся данных. Дополнение и редактирование данных происходят аналогично с текстовым редактором. Сортировка имеет два ключа: сортировать (для чисел) в порядке возрастания номеров;

сортировать (для чисел) в порядке убывания номеров;

Таблица 2

1	2	3	4	5	6
	Фамилия	Имя	Физика	Химия	ОИВТ
1	Авакумов	Эдуард	5	5	5
2	Алексеева	Ольга	3	3	3
3	Захаров	Игорь	4	3	3
4	Смирнова	Ольга	4	5	4
5	Соколов	Игорь	3	3	4
6	Титов	Николай	4	5	5
7	Хололов	Алексей	3	4	4
8					

сортировать (для текста) по алфавиту; сортировать (для текста) в порядке, обратном алфавиту.

Ключ задается в столбце, в котором пользователь производит действие.

Зададим ключ *сортировать в порядке убывания номеров* и выполним команду сортировки (табл. 3).

Таблица 3

1	2	3	4	5	6
	Фамилия	Имя	Физика	Химия	ОИВТ
8					
7	Хололов	Алексей	3	4	4
6	Титов	Николай	4	5	5
5	Соколов	Игорь	3	3	4
4	Смирнова	Ольга	4	5	4
3	Захаров	Игорь	4	3	3
2	Алексеева	Ольга	3	3	3
1	Авакумов	Эдуард	5	5	5

Приведем теперь пример выполнения команды поиска. Зададим в качестве ключа поиска имя «Эдуард» в соответствующем поле базы. Результат выполнения команды показан на табл. 4.

Примеры использования для поиска других ключей: табл. 5 — выбраны все записи, имеющие 4 в четвертом поле, табл. 6 — содержимое пятого поля больше трех.

6-й урок. На этом занятии целесообразно дать учащимся, которые уже знакомы с учебной базой данных, выполнить большое самостоятельное задание.

Таблица 4

1	2	3	4	5	6
	Фамилия	Имя	Физика	Химия	ОИВТ
1	Авакумов	Эдуард	5	5	5

Таблица 5

1	2	3	$x=4$	5	6
	Фамилия	Имя	Физика	Химия	ОИВТ
2	Захаров	Игорь	4	3	3
3	Смирнова	Ольга	4	5	4
4	Титов	Николай	4	5	5

Таблица 6

1	2	3	4	$x>3$	6
	Фамилия	Имя	Физика	Химия	ОИВТ
1	Авакумов	Эдуард	5	5	5
2	Смирнова	Ольга	4	5	4
4	Титов	Николай	3	4	4
5	Хололов	Алексей	3	4	4

Задание. Сформируйте базу данных «Класс», в которую поместите такие данные: фамилия, имя, отчество учащихся, год рождения, домашний адрес и телефон.

Работа с учебными пакетами прикладных программ

Цель. Дать понятие о пакете прикладных программ; познакомить учащихся с возможностями использования компьютеров для решения задач из различных областей науки и практики.

Требования к знаниям и умениям. Учащиеся должны знать назначение и состав учебного пакета прикладных программ, уметь использовать любую программу пакета для решения конкретной задачи из соответствующей области.

Методические указания. Для работы с учебными пакетами программой практикума предусмотрено 3 учебных часа. В связи с тем что состав пакетов для КУВТ-86 и КУВТ «Ямаха» существенно различен, в данном разделе предлагается два варианта организации практических занятий.

Рекомендуется следующее примерное планирование.

7-й урок. Работа с электронными таблицами.

Электронные таблицы представляют собой специальный вид прикладного программного обеспечения, ориентированный в основном на экономические

расчеты. Они организованы таким образом, что изменение каких-либо элементов таблицы немедленно влечет за собой автоматический пересчет всех связанных с этими элементами значений. Это позволяет быстро видеть результаты корректировки исходных данных и анализировать их.

Для работы с электронными таблицами на КУВТ-86 учащимся предлагается меню из пяти режимов:

- создание таблицы;
- редактирование;
- добавление данных;
- выход в меню;
- окончание работы.

Шаблоны таблиц здесь заготовлены заранее. Ученику достаточно лишь выбрать нужный.

После ввода режима на экране дисплея появляется выбранная электронная таблица. В нижней части экрана выдается дополнительная информация — сообщения программы об ожидаемых действиях пользователя.

Пример. Составление сметы туристского похода.

Воспользуемся соответствующей электронной таблицей, выбрав из меню нужный режим. Шаблон таблицы изображен на табл. 7.

Вначале необходимо внести в таблицу данные. Занесем слово «рюкзак» в столбец «Название». Для этого необходимо указать номер формируемой строки. Номера строк заносятся в таблицу, курсор автоматически устанавливается в ячейку «Название» соответствующей строки. Следует учитывать, что длины полей фиксированы: одно — длиной 11 символов при строке экрана 64 и 8 при строке 32 символа, два — длиной соответственно 5 и 3, три — длиной 3. Если длина названия меньше, оставшиеся позиции заполняются пробелами или клавишей →.

Таблица 7

Номер	Название	Цена	Количество	Стоимость

Набрав слово «рюкзак», нужно ввести данные о цене и количестве, которые автоматически заносятся в соответствующие графы. Вводимые данные не должны превышать длину поля таблицы. Стоимость вычисляется автоматически и заносится в последнюю ячейку строки.

Если продолжить заполнение таблицы, то в графе «Номер» появится номер следующей строки, и т. д. После прекращения заполнения таблицы происходит автоматический подсчет и в нижнюю часть заносится итог (табл. 8).

Таблица 8

Номер	Название	Цена	Количество	Стоимость
1	Рюкзак	43	4	172
2	Палатка	190	1	190
3	Спальный мешок	55	4	220
4	Байдарка	236	2	472
5	Котлы	10	1	10
ИТОГО			12	1064

Учебные электронные таблицы, включенные в состав пакета для КУВТ «Ямаха», несколько отличаются от описанных. Помимо числовых данных они позволяют задавать и формулы, связывающие отдельные элементы таблиц.

Шаблон таблицы изображен на табл. 9.

В нижней части экрана высвечена справочная информация — назначение функциональных клавиш. Пользуясь ими, можно осуществлять ввод, корректирование информации, задание связей между элементами (с помощью формул) и т. д.

Следует заметить, что набор допустимых формул ограничен: можно использовать лишь арифметические опе-

Таблица 9

	A	B	C
1			
2			
3			
4			
5			

рации — сложение, вычитание, умножение и деление, притом не более чем над двумя аргументами, а также суммирование элементов таблицы по столбцам и по строкам.

Пример: задание формулы $C2 = A1 + B1$ означает, что в ячейке C2 таблицы всегда будет находиться сумма двух первых элементов верхней строки таблицы.

Формула $B1 = \text{sum}(A1:A5)$ означает, что в ячейку B1 будет заноситься сумма первых пяти элементов первого столбца.

Если мы хотим использовать более сложные зависимости, нужно задавать их последовательно, например:

$$B1 = A1 + A2$$

$$C1 = B1 / 2 \text{ и т. д.}$$

Задание 1. С помощью электронной таблицы подсчитать сумму квадратов 20 чисел.

Задание 2. Составить с помощью электронной таблицы смету семейного бюджета.

8-й урок. Работа с графиками функций.

Следует отметить, что современные ПЭВМ, как правило, снабжены богатыми библиотеками программ, предназначенных для научно-исследовательской работы. Примером могут служить учебные программы для исследования графиков функций.

Рассмотрим возможности анализа графиков функций на КУВТ-86.

Программа начинается с меню из пяти пунктов:

задание функций;

построение графиков;

подсчет значений функций для заданных x ;

работа с магнитофоном;

выход из программы.

При выборе пункта 1 программа в диалоговом режиме позволяет задать начало координат, масштаб по осям X и Y, аналитическую запись исследуемых функций. Одновременно можно задать три функции, например:

$$y = F \sin(x)$$

$$y = F \cos(x)$$

$$y = 0.5 \times x$$

Функции могут быть заменены с по-

мощью клавиш редактирования. Отредактированные функции засылаются в память нажатием клавиш «ввод, G, ввод». Для исследования одной функции следует две другие положить равными нулю (или ввести все три одинаково). Кроме того, возможно задание отрезков, на которых функции не определены.

Возвратившись в главное меню, можно теперь выбрать режим построения графиков (пункт 2).

После того как графики будут построены на экране дисплея, программа предоставляет следующие возможности для их исследования:

подогнав курсор к интересующей нас точке, узнать ее координаты (нажав клавишу «пробел»);

узнать значения функций для заданного значения аргумента x (нажав клавишу «ввод», выйти в меню, после чего, выбрав пункт 3, задать значение x).

Кроме того, программа позволяет обращаться к внешнему запоминающему устройству — магнитофону и:

записать программу на ленту;

записать значения переменных на ленту;

читать значения переменных с ленты в память.

Программа исследования графиков, входящая в состав пакета для КУВТ «Ямаха», также позволяет производить указанные действия (за исключением работы с магнитофоном). Кроме того, она позволяет менять интервал исследования функции, автоматически масштабирует график по вертикали, исследуемую функцию можно задавать в более общем виде.

У п р а ж н е н и е. Решить с помощью программы исследования графиков функций уравнение $x + 1 = \exp(x)$.

У п р а ж н е н и е. Построить график функции $y = x^3 + 0.95 - 1/(x^2 + 1)$ на отрезках $[-2; 2]$; $[-0.1; 0.1]$; $[-1; 1]$ и исследовать функцию на монотонность.

У п р а ж н е н и е. Найти нули следующих функций:

а) $y = x^5 + x^3 + x - 20$;

б) $y = x^7 - 4x^6 + 2x^5 - 3x^4 + 6x - 1$.

В случае а) следует правильно выбрать интервал построения графика.

В случае б) функцию нужно рассматривать на отрезке $[-4; 4]$, иначе можно не найти третий нуль функции.

9-й урок. Работа с учебными программами по геометрии, физике, истории и другим предметам.

В состав учебных пакетов прикладных программ входит целый ряд обучающих программ по большинству предметов школьного курса. Познакомиться со всеми из них учащиеся, вероятно, не успеют, поэтому целесообразно сосредоточить внимание на 1—2 типичных программах. Этого достаточно, чтобы дать ученикам представление о принципах их построения и работы.

В качестве примера опишем работу с двумя программами по географии из учебного пакета для КУВТ-86.

Первая программа предназначена для контроля с элементами обучения. В ней организована работа с картой Советского Союза в диалоговом режиме с подсказками. Работа начинается с сообщения на экране: «У карты СССР». Следует ввести название столицы, после чего на экране появляется карта Советского Союза, по которой необходимо определить названия зашифрованных буквами городов. Города обозначены буквами алфавита от А до Ю. Условные обозначения городов на карте печатаются на экране после заставки. Нажав клавишу с точкой, можно вызвать эту подсказку в любой момент.

Далее программа начинает задавать вопросы: «Какой город на карте обозначен буквой...?» Посмотрев на экран, нужно набрать название этого города заглавными русскими буквами. В конце программы выдается число ошибочных ответов. Нажатие клавиши «ввод» обеспечивает переход к началу программы.

Это характерный представитель учебных программ-тренажеров, которые нацелены на отработку узкого раздела изучаемого курса.

Вторая программа сообщает сведения о государствах Европы. Она представляет собой специализированную базу данных, каждая запись которой разделена на четыре поля, несущих определенную информацию: 1-е — название государства (из 28 символов); 2-е — название столицы (20 символов); 3-е —

население страны (28 символов); 4-е — площадь страны (8 символов). С помощью этой базы данных можно получать справочную информацию. Запросы сопровождаются двумя символами «,?».

Пример: просмотрим записи о государствах, где живут немцы и французы. Запрос набирается так:
нем франц,?

Пробел между словами обозначает соединительный союз «и», запятая между словами — союз «или». Если требуется информация о государствах, в которых живут немцы или французы, запрос должен иметь вид
нем, франц,?

Чтобы получить записи о каком-либо государстве, нужно набрать название этого государства, например:

24 Франц,?

Будут выданы все записи, содержащие слово «франц».

Записи в базе данных можно изменять. Для этого на вопрос: «Хотите изменить запись?» — следует ответить «да».

Для исключения записи в режиме коррекции нужно вместо названия страны ввести символ *.

Для ввода новой записи следует найти пустую запись (содержащую в первом поле символ *) и заполнить ее в режиме коррекции.

У п р а ж н е н и е . Выполнить следующие задания:

- 1) найти страну со столицей Вадуж;
- 2) найти страну площадью 551 кв. км;
- 3) исключить из базы страну Монако.

В составе пакета для КУВТ «Ямаха» есть деловая игра по экономической географии «Инфраструктура».

На экране возникает условная контурная карта некоего района с отмеченными населенными пунктами, путями сообщения, источниками полезных ископаемых и энергии. В нижней части экрана приведены условные обозначения (цветными кружками) различных предприятий — алюминиевый комбинат,

автозавод и т. д. В левой верхней части экрана последовательно появляются эти значки. Клавишами управления курсора нужно подгонять их к изображениям населенных пунктов и совмещать.

После размещения всех предприятий программа анализирует его на оптимальность и оценивает решение в баллах.

У п р а ж н е н и е . Выполнить задание, предлагаемое программой.

Хороший пример инструментального учебного средства — программа по геометрии для КУВТ «Ямаха». Она реализует на компьютере основные геометрические построения, записанные на специальном языке графических команд.

После запуска программы на экране дисплея появляется вопрос: «Нужна ли справочная информация? В случае ответа «да» выдается список графических команд и описание действия каждой из них. Команды включают как простейшие геометрические построения (поставить точку, провести прямую, окружность и т. д.), так и дополнительные (найти середину отрезка, провести биссектрису угла и др.). Затем ученику предлагается ввести «программу» — последовательность графических команд, осуществляющую какое-либо геометрическое построение.

Предусмотрен синтаксический анализ команд: в случае ошибочной записи появляется сообщение об ошибке. Признаком окончания программы служит команда «стоп» (или просто «с»).

После того как программа ученика введена, на экране последовательно выполняются геометрические построения, соответствующие введенным командам.

У п р а ж н е н и е . С помощью графической системы выполнить следующие построения:

- а) построить треугольник ABC и найти точку пересечения его медиан;
- б) описать вокруг треугольника ABC окружность.

Завершение практических занятий

Четвертый раздел практикума завершается 2-часовой экскурсией на вычислительный центр или в дисплейный класс.

Экскурсия — форма организации обучения, позволяющая провести наблюдения, расширить познания в изучаемой области, увидеть связь теоретических знаний с практическим их применением, предоставляющая широкие возможности осуществления межпредметных связей, способствующая политехническому обучению и профессиональной ориентации учащихся. Она должна проводиться с одновременным изучением на уроках определенных разделов программы.

Проведение экскурсии требует тщательной подготовки. Учитель должен определить ее задачи и содержание, выбрать вычислительный центр и ознакомиться с ним, уяснить для себя все сложные моменты, которые могут возникнуть.

Заблаговременно ученики должны получить задания, где необходимо указать, какие наблюдения должен провести каждый из них, на какие вопросы он должен дать самостоятельные ответы, какие материалы собрать, в какой форме (устное сообщение, реферат, коллекция, альбом и т. п.), к какому сроку подготовить отчет об экскурсии.

Важно обеспечить активную работу всех учащихся, поддержание дисциплины и порядка, знание и соблюдение правил техники безопасности.

Заключительный этап экскурсии включает проведение учителем итоговой беседы, в ходе которой полученные на экскурсии сведения включаются в общую систему знаний учащихся. Материалы экскурсии можно использовать для организации выставки, проведения специального занятия, в качестве раздаточного материала на соответствующих уроках.

Если экскурсию будет проводить не сам учитель, то с учащимися нужно провести дополнительное занятие для разъяснения содержания экскурсии, а сотрудника ВЦ познакомить с содержи-

ем экскурсии, ее целями и задачами, определить с ним правила поведения учащихся на вычислительном центре.

Предложим несколько тем экскурсии.

Прикладное программное обеспечение. *Цель.* Познакомить учащихся с имеющимся программным обеспечением в дисплейном классе.

Содержание. Необходимо акцентировать внимание учащихся на классификации программ, каждый класс которых предназначен для решения конкретного типа задач.

К самым распространенным классам прикладных программ можно отнести следующие:

1. Электронные таблицы. Это своеобразные «калькуляторы», позволяющие гибко манипулировать информацией и производить различные вычисления.

2. Текстовые редакторы. Предназначены для разнообразной редакторской работы с текстами.

3. Базы данных. Предназначены для хранения большого количества информации и быстрого ее поиска.

4. Графические редакторы. Предназначены для построения и редактирования различных рисунков и чертежей.

5. Интегрированные системы. Объединяют в единое целое несколько прикладных программ. Предназначаются для решения различных типов задач.

6. Тренажеры, игровые программы и др. Следует отметить, что программное обеспечение для компьютеров — это тысячи наименований прикладных программ различного назначения. Поэтому учителю (совместно с работниками ВЦ) необходимо заранее определить круг программ, которые позволят наиболее полно продемонстрировать сферы их применения. При отборе необходимо помнить, что очень часто на рынок программного обеспечения попадают программы низкого профессионального уровня.

Демонстрацию программного обеспечения необходимо связать с практиче-

скими занятиями, которые проводились с учащимися.

Применение ЭВМ в народном хозяйстве. *Цель.* Познакомить учащихся с применением ЭВМ в народном хозяйстве, с социальными последствиями применения ЭВМ.

Содержание. Разработка и внедрение более эффективных форм управления производством — одна из важнейших задач, поставленных партией в целях ускорения социально-экономического развития страны на базе научно-технического прогресса.

Важнейшим этапом управления является получение своевременной, полной и объективной информации о ходе того или иного процесса. Стремительный рост масштабов современного производства привел к тому, что системы административного управления все чаще стали давать сбои, т. е. запаздывать с принятием решений, так как перестали справляться с возрастающим потоком информации, на основе которой эти решения принимаются. Вот почему в настоящее время на предприятиях проделана значительная работа по внедрению АСУП на базе вычислительных центров предприятий.

Примером может служить Волжский автомобильный завод в г. Тольятти.

ВАЗ ежегодно выпускает около миллиона легковых автомобилей. Сборочный конвейер в главном корпусе этого завода работает под управлением ЭВМ. Ее задача — следить за своевременным поступлением деталей на конвейер со складов и цехов вспомогательных производств. Для этого соответствующая программа собирает и перерабатывает сведения о тысячах деталей, образующих автомобиль, о наличии деталей на складах, об их движении по транспортным линиям, о поступлении на рабочие места конвейера.

Интересные результаты получены на сельскохозяйственных предприятиях Гомельской области. Распределение обязанностей по производству и выращиванию молодняка, производству кормов, заключению договоров с другими производителями, заключительной фазе откорма — все это было сначала предметом тщательного экономического анализа

(имеется в виду построение экономико-математической модели и ее анализ на ЭВМ). Затем, когда перспектива замысла стала очевидной, был начат эксперимент. Его эффект превзошел все ожидания. В головном хозяйстве производство мяса выросло за 4 года почти в 3 раза, расход кормов на единицу привеса уменьшился с 9,4 кормовых единицы до 7. Урожайность кормовых культур возросла с 24 до 60 ц с гектара, а трудовые затраты на 1 ц сократились с 23,7 человеко-часа до 7,6. Себестоимость центнера привеса снизилась со 170 до 117 руб.

Подготовку подробных примеров учитель может поручить учащимся.

В процессе экскурсии учащиеся должны увидеть одно из частных использований ЭВМ в народном хозяйстве.

Общая схема ЭВМ. *Цель.* Рассказать учащимся об общей схеме ЭВМ, общих принципах организации хранения информации в памяти ЭВМ и обмене информацией между ее устройствами.

Содержание. Конструкции, размеры, внешний вид, быстрдействие ЭВМ и формы взаимодействия их с человеком непрерывно меняются, одно поколение машин уступает место другому. Однако изменения не коснулись основных принципов работы и основных устройств ЭВМ. Всякая вычислительная машина с программным управлением состоит из следующих блоков (устройств):

- ввода информации;
- вывода информации;
- управления;
- арифметическо-логического устройства (процессора);
- оперативного запоминающего устройства (ОЗУ);
- внешнего запоминающего устройства.

Устройство ввода служит для занесения в ОЗУ текста программы, руководствуясь которой ЭВМ будет решать задачу, а также исходных данных. Современные ЭВМ снабжаются различными типами устройств ввода. Если информация цифровая или буквенная, то для ее ввода используется клавиатура. Скорость ввода информации с помощью клавиатуры ограничена возможностями человека — 200 символов в минуту. Если необходимо ввести в память графиче-

Решение учебных задач с использованием ЭВМ

скую информацию — чертеж или рисунок, то применяются панели графического ввода с контактными карандашами или так называемые «световые перья». Совершенствуются системы для ввода в память ЭВМ информации «под диктовку» человека.

Большое распространение получили устройства ввода, в которых информация считывается с перфолент или перфокарт. Достоинства таких систем в том, что перфолента и перфокарты могут быть использованы многократно, считывание информации с них осуществляется с большой скоростью. Устройство ввода может «прочитать» за минуту до 770 перфокарт. С перфоленты за секунду считывается 100 символов.

Устройство управления обеспечивает координацию, согласовывает взаимодействие блоков ЭВМ в соответствии с программой.

В процессоре выполняются все арифметические и логические операции над данными, представленными в двоичных кодах. Современные процессоры работают со скоростью до сотен миллионов операций в секунду.

ОЗУ служит для хранения выполняемой программы и основной части данных.

К внешним запоминающим устройствам относятся магнитные барабаны, магнитные диски, магнитофоны. На них можно записать большие объемы информации и иметь достаточно быстрый доступ к ней.

Для вывода информации применяются алфавитно-цифровые печатающие устройства, монитор, графопостроитель. По окончании работы ЭВМ выводит результаты на бумагу или экран монитора.

Рассказ об основных устройствах должен сопровождаться их демонстрацией на конкретной ЭВМ.

По своему усмотрению учитель может выбрать одну тему или соединить в одну несколько тем экскурсии, исходя из конкретной ситуации и имеющегося времени.

Здесь нужно продемонстрировать учащимся возможности решения различного рода задач из курсов физики, математики и других предметов с использованием ЭВМ.

На решение учебных задач с использованием ЭВМ программой практических занятий отводится 4 учебных часа. Рекомендуется следующая организация практикума:

2 ч — решение задач по физике и математике;

2 ч — решение задач по другим предметам.

При проведении практических занятий необходимо заранее подобрать круг задач. Задачи должны быть посильны для всего класса. Можно подобрать задачи по нескольким группам сложности, соответствующим силе учащихся.

Решение задач можно разделить на два типа:

с применением инструментального средства;

с самостоятельным составлением программ.

Интересны и познавательны задачи по математике на построение графиков функций (с применением инструментальных средств). Ученикам могут быть предложены следующие виды задач:

исследовать поведение функции при изменении коэффициентов;

исследовать поведение функции на определенном интервале;

решить систему двух уравнений графическим способом.

Инструментальная программа построения графиков функций позволяет организовать решение задач исследовательского характера. Построены и исследованы могут быть практически любые графики. Наглядно может быть проиллюстрирован графический метод решения системы уравнений, изучение которого в курсе школьной математики связано с трудностью выбора классов функций.

При решении физических задач можно воспользоваться существующими инструментальными программами-практикумами. Они позволяют решать задачи

экспериментально, исследовательским путем. Постановка конкретной задачи будет зависеть от имеющихся программных средств.

Круг задач второго типа также разнообразен. Можно предложить учащимся следующие:

написать программу для вычисления объема (площади) фигуры;

написать программу для приближенного вычисления квадратного корня (или числа «пи»);

написать программу для нахождения интегральной суммы;

определить высоту подъема тела, брошенного вертикально (под углом к горизонту);

вычислить скорость, набранную телом, движущимся равноускоренно;

28 вычислить объем газа при заданных условиях;

вычислить силу взаимодействия электрических зарядов;

рассчитать полное сопротивление электрической цепи (при параллельном, последовательном, смешанном соединениях);

рассчитать фокусное расстояние линзы.

Пример 1. Вычислить квадратный корень с заданной точностью.

Решение задачи при помощи ЭВМ можно разделить на следующие этапы: постановка задачи;

выделение величин в задаче;

математическая модель задачи;

составление алгоритма решения задачи;

составление программы решения задачи;

тестирование программы.

Постановка задачи. Необходимо составить программу, по которой можно вычислять квадратный корень с определенной степенью точности.

Величины задачи. X — исходное число; Y — результат вычисления; E — заданная точность вычисления.

Математическая модель.

$Y_1 = (X + X/X) / 2$ — первое уточнение корня;

$Y_2 = (Y_1 + X/Y_1) / 2$ — второе уточнение корня;

$Y_3 = (Y_2 + X/Y_2) / 2$ — третье уточнение корня;

...

Процесс будет выполняться до тех пор, пока $|(Y_n)^2 - X| \geq E$.

Алгоритм задачи.

алг Квадратный корень (вещ X, Y, E)

арг X, E

рез Y

нач $Y := X$

пока $|Y^2 - X| \geq E$

нц

$Y := (Y + X/Y) / 2$

кц

кон

Программа решения задачи (язык Бейсик).

10 REM Уточнение квадратного корня

20 CLS

30 PRINT 'Нахождение значения квадратного корня с заданной точностью'

40 PRINT:PRINT 'Введите число':;

INPUT X

50 PRINT:PRINT 'Укажите степень точности': INPUT E

60 $Y = X$

70 IF ABS ($Y * Y - X$) >= E THEN $Y = (Y + X/Y) / 2$:GOTO 70

80 PRINT:PRINT 'Значение квадратного корня из 'X' равно 'Y

90 END

Тестирование программы.

Оно проводится для тех значений аргументов, при которых результат можно посчитать без ЭВМ. Протестируем нашу программу для $X=9$ с точностью 0,0000000001. Получим результат 3.

Более сильным ученикам можно предложить усовершенствовать программу: ввести ограничение на область допустимых значений аргумента; осуществить вывод номера шага уточнения и т. д.

Пример 2. Найти путь, пройденный автомобилем при равноускоренном движении, если известны начальная скорость, время движения и ускорение, с которым движется автомобиль.

Программа решения задачи.

10 REM Нахождение пути, пройденного телом

20 CLS

30 PRINT 'Нахождение пройденного пути'

40 PRINT:PRINT 'Введите значение начальной скорости автомобиля (м/с)':; INPUT V0

50 PRINT:PRINT 'Введите значение

```

ускорения (м/с2)'; INPUT A
60 PRINT: PRINT 'Введите время дви-
жения (с)'; INPUT T
70 S=V0*T+A*T*T/2
80 PRINT: PRINT 'Автомобиль прошел
путь' S' м'
90 END

```

Дополнительные задания. Выводить значение пути в определенном формате; вывести сообщение о направлении движения, вывести скорость в конце пути и т. п.

Пример 3. Рассчитать максимальную дальность полета снаряда и высоту подъема.

Программа.

```

10 REM Движение тела под углом к
горизонту
20 CLS
30 PRINT 'Расчет максимальной даль-
ности полета и максимальной высоты
подъема снаряда'
40 PRINT: PRINT 'Введите значение
начальной скорости снаряда
(м/с)'; INPUT V0
50 PRINT: PRINT 'Введите значение
угла наклона орудия к горизонту
(град)'; INPUT A
60 H=V0*V0*SIN(A*3.14/180)
/(2*9.81)
70 S=V0*V0*SIN(2*A*3.14/180)/9.81
80 PRINT: PRINT 'Снаряд пролетит
расстояние 'S'м'
90 PRINT: PRINT 'Максимальная вы-
сота подъема 'H'м'
100 END

```

Дополнительные задания. Выводить значение высоты и дальности полета в определенном формате; вывести сообщение о времени движения снаряда; ввести ограничение для значения на-

чальной скорости и угла наклона орудия и т. п.

Ученикам, которые занимаются информатикой дополнительно, можно предложить построить модель конкретного физического явления.

Следует помнить, что задачи не должны изобилывать сложными математическими выводами и выкладками. Каждый из учеников должен к концу занятия получить конкретное решение.

Для решения задач по другим предметам в пакетах прикладных программ также имеются инструментальные средства, которые учитель может применить на следующих двух уроках.

Задачи широкого класса (практически по всем предметам) могут быть решены при использовании логических языков, таких, как Пролог. Он реализован на некоторых ПЭВМ. Начальное знакомство с этим языком практически не требует времени, но даже его хватает, чтобы решать довольно сложные задачи.

В конце практикума ученикам можно предложить составить программу контроля знаний по нескольким вопросам конкретного предмета. Такое задание обычно вызывает у учащихся живой интерес. На уроке можно дать им возможность обмениваться программами; обычно при этом учащиеся осознают все достоинства и недостатки своих программ, самостоятельно начинают совершенствовать алгоритм. При решении этой несложной задачи перед учениками раскрываются многие подводные камни, лежащие на пути решения задач с применением ЭВМ; они более точно осознают необходимость знаний, полученных на теоретических занятиях, и возможность их применения на практике.

29

Издательство «ПЕДАГОГИКА», одним из подразделений которого является и наш журнал, перечислило премию, полученную коллективом за 1 место в социалистическом соревновании по итогам III квартала, в Детский фонд имени В. И. Ленина. Мы хотим, чтобы эти деньги были вручены выпускникам одного из детских домов.

О двух последних темах X класса

О роли ЭВМ в обществе

30

Применение ЭВМ в самых различных, иногда довольно экзотических, областях освещается сегодня широко, на недостаток иллюстративного материала учителю жаловаться не приходится. Возьмите недельный комплект любой центральной газеты — и вы наверняка найдете там 2—3 заметки с заголовками типа «На помощь приходит ЭВМ» или «Умные помощники человека». К сожалению, при чтении некоторых статей создается впечатление, что с появлением ЭВМ многие проблемы решаются сами собой. Авторы газетных дифирамбов редко упоминают о том, что конкретную пользу приносит в конечном счете не сама машина, а программа, под управлением которой она работает, не вспоминают они и о таком важнейшем этапе, как постановка задачи и ее формализация.

Казалось бы, эти умолчания вполне оправданны. Газета издается не для специалистов, ее задача — формирование широкого общественного мнения, и сегодня, когда необходимо выработать в обществе позитивный настрой в отношении к новой технике, хвалебные заметки представляются вполне уместными. Но... только на первый взгляд.

Люди, поверившие во всемогущество ЭВМ, испытывают жестокое разочарование, столкнувшись с ней в жизни. Выясняется, что «мыслящая машина» не умеет работать самостоятельно, что она не облегчает, а, скорее, интенсифицирует умственный труд человека, предъявляя к нему качественно новые требования. Люди оказываются не готовы к этому. Отсюда — крах многочисленных АСУ, неприязненное отношение к вычислительной технике, протесты против введения курса ОИВТ.

Таким образом, фетишизация компьютера и страх перед ним — две стороны одной медали, одинаково далекие от истины. Им необходимо противопоставить реальные представления, подлинную компьютерную грамотность. По-

этому, говоря об использовании ЭВМ, не стоит ограничиваться рассказами о триумфальном шествии вычислительной техники, о ее действительно необъятных возможностях. Необходимо упомянуть и о проблемах, причем не только «машинных» (технических), но и о «человеческих» (социальных). К сожалению, часов, которые отводятся сейчас на эту тему, явно недостаточно, но можно надеяться, что в ближайшем будущем эта несправедливость будет исправлена, а пока нарисуем картину в самых общих чертах.

ЭВМ предназначена для решения задач, но у нее нет разума, она не может ни выбрать задачу, ни определить метод ее решения. Это должен сделать человек. Так на заре развития вычислительной техники между машиной и задачей встали два человека: специалист и программист.

Специалист — это инженер, врач, ученый, попросту говоря, человек, который знает, какую задачу надо решать, и сумеет использовать результаты решения.

Для общения с машиной необходимо было в деталях знать ее устройство, что, разумеется, требовало длительной подготовки. Между Специалистом и машиной возникает посредник — Программист. Программист тех времен напоминает античного жреца: только он может передать просьбу простого смертного (специалиста) божеству (машине), получить ответ и истолковать его. Фигура Программиста трагически противоречива. Только ему подчиняется машина, но он вынужден подавать ей команды с чужого голоса, он должен решать задачи из чуждой ему предметной области.

Все действующие лица связаны в очень простую цепочку: задача — Специалист — Программист — машина.

В те далекие времена эта схема работала вполне удовлетворительно. Задачи решались сравнительно несложные, в основном вычислительные, не связанные с большими объемами сложно организованных данных. Большого не поз-

воляли ни возможности машин, ни знания специалистов.

Машины накладывали количественные ограничения: у них было низкое быстродействие, малая память, не развиты средства ввода—вывода.

Качественные ограничения на решаемые задачи возникали из-за неполноты представлений специалистов. Подлинными возможностями компьютера еще не осознавались, он воспринимался лишь как средство, позволяющее быстро и точно выполнить большой объем вычислений. О принципиально новых задачах и методах решения, не реализуемых без помощи компьютера, только начинали задумываться. Пока что на ЭВМ переносились привычные ручные методы обработки или их модификации, отличающиеся количественно, но не качественно. Еще одно ограничение на сложность задачи проистекало из того, что она должна была быть понятна Программисту.

Заметим, что структура этой цепочки основывалась на том, что проще было обучить Программиста специальности, чем Специалиста программированию. Вспомним, например, персонажей написанной в начале 60-х гг. знаменитой сказки братьев Стругацких «Понедельник начинается в субботу»: программист Саша Привалов умеет обращаться с «Алданом», магистры Роман Ойра-Ойра и Эдик Амперян владеют магией. И маги-магистры охотно берут ничего не смыслящего в чародействе Сашу на работу в НИИЧАВО, потому что им проще обучить его основам волшебства, чем самим разбираться в вычислительной машине.

Шло время. Машины развивались. Их возможности постепенно перерастали предлагаемые им задачи. В то же время люди стали осознавать, какой мощный инструмент находится в их руках, стали вырисовываться принципиально новые, очень сложные задачи.

Машины и задачи были готовы к скачку, к переходу на какой-то новый уровень. Не готовы оказались люди. От Специалиста, привыкшего просто передавать машине часть своей работы (или делать все самому, не желая знать никаких компьютеров), вдруг потребова-

лось по-новому взглянуть на всю свою деятельность, пересмотреть круг решаемых задач, выработать информационный подход к ним. И дело тут не в том, что необходимо загрузить машину работой. Просто старые решения старых задач перестали давать удовлетворительные в новых условиях результаты, а Специалист оказался не готов к этому. Он по-прежнему пытался обойтись без машины или возложить на нее какие-то простейшие функции, а дело не клеилось, и часто оказывалось, что удобнее всего обвинять в этом машину и Программиста.

Но и Программисту было не легче. Новые задачи потребовали глубокого знания предмета, которым он не обладал и без которого до этого обходился. Разбираясь в ЭВМ, но не зная предметной области, Программист оказался в положении человека, в совершенстве владеющего сложным инструментом, но не знающего, где его применить. Трудно, почти невозможно стало находить общий язык со Специалистом. Многочисленные Саши Приваловы перестали справляться со своей работой.

...Все эти печальные картины описаны в прошедшем времени, хотя кризис не преодолен и сегодня. Но выход есть. Его подсказывает машина.

Обратная сторона сложности — простота. Невероятно усложненная, с тысячекратно возросшими возможностями, машина стала помещаться на письменном столе, а управление ею упростилось настолько, что с ним справляются дети. И если раньше проще было Программисту изучить основы специальной предметной области, чем Специалисту разбираться в ЭВМ и программировании, то теперь ситуация меняется. Усложнение задач делает их недоступными для Программиста, поэтому необходимой становится самостоятельная работа Специалиста с ЭВМ. С другой стороны, упрощение работы с машиной обеспечивает эту возможность.

Так возникает необходимость компьютерной грамотности Специалиста, а следовательно, всеобщей компьютерной грамотности, ибо каждый человек — это специалист в своей профессии, в своей сфере деятельности. И требуется не

просто представление об ЭВМ и умение нажимать на клавиши. Нужно уметь по-новому взглянуть на свою специальность, выявить ее информационное содержание, сформулировать соответствующие задачи, определить, какие из них целесообразно решать с помощью компьютера. Нужна не просто компьютерная, нужна информационная грамотность.

Все эти обстоятельства сделали необходимым введение курса основ информатики и вычислительной техники в среднюю школу. Информационная грамотность — вот глобальная задача курса. И обсуждая тему «Роль ЭВМ в современном обществе», очень важно взглянуть на курс информатики в целом, показать школьникам причины появления этого предмета в их расписании.

32

Итак, мы пришли к выводу, что Специалист должен научиться по-новому ставить задачи и самостоятельно решать их на ЭВМ. Таким образом, он перестает нуждаться в посреднических услугах Программиста. Следует ли отсюда, что Программист должен исчезнуть? Да, если речь идет о прикладном программисте, который появился на сцене, чтобы помочь Специалисту найти общий язык с машиной; нет, если иметь в виду системного программиста.

Программист не исчезает вообще, он просто меняет роль. Новая схема взаимодействия выглядит так: задача — Специалист — машина — Программист.

Программист становится невидим Специалисту, он спрятан от него за машиной. Сфера его деятельности — системные программы: операционные системы, системы программирования, системы управления базами данных, сервисные системы и т. д. Именно они обеспечивают Специалисту комфорт при общении с машиной.

К сожалению, машина, разделяющая Специалиста и Программиста, часто становится непреодолимой стеной непонимания, а порой и баррикадой. Необходимо, чтобы лишенные прямой связи Специалист и Программист сохраняли взаимопонимание, чтобы каждый из них правильно представлял свою роль в общем деле. Нельзя забывать, что за

столь привлекательной для Специалиста внешней простотой работы с машиной стоит большой труд Программиста, а с другой стороны, реальную пользу приносит решение прикладных задач, поэтому труд Программиста имеет смысл, только если Специалист будет пользоваться его результатами.

Основы такого понимания проблем, связанных с ЭВМ и человеком, могут (и должны!) закладываться на школьных уроках информатики.

Об устройстве ЭВМ

Цель изучения этого раздела так сформулирована в книге для учителя: «...показать принципиальную возможность организации автоматического исполнения компьютером программы, подчеркнув при этом, что компьютер не более чем сложная схема». Учебное пособие построено в точном соответствии с этой целью: в нем доказывается, что автоматическое исполнение алгоритмов возможно, для чего рассматриваются принципы работы конкретной ЭВМ.

Однако такой подход представляется неполным. Одна из глобальных целей курса — формирование представлений о реальных возможностях современных компьютеров. Это требует проведения в разделе «Устройство ЭВМ» функционального подхода к компьютеру: подробнее изучить отдельные устройства, их характеристики (внешние характеристики, а не детали внутренних принципов работы), связанные с ними возможности ЭВМ в целом, взаимосвязи устройств.

Здесь будет дана попытка подобного взгляда. Необходимо сразу оговориться: хотя о программах не будет ни слова, указанные возможности могут превратиться в действительность только при наличии соответствующего программно-го обеспечения.

Общая схема ЭВМ. Прежде чем говорить об устройстве компьютера, необходимо уяснить суть этого прибора. Бытующее мнение, что компьютер — это большое вычислительное устройство, некий свехарифмометр или суперкалькулятор, в корне неверно. Отличие ЭВМ от калькулятора не количествен-

ное, а качественное. Название «электронно-вычислительная машина» (как и «компьютер», означающее «вычислитель») сложилось исторически и не отражает сущности предмета.

Компьютер — это устройство для переработки произвольной (а не только числовой!) информации; это информационная машина.

Понимание информационной сущности компьютера объясняет феномен его всеобщей применимости. В самом деле, в истории человечества не было еще технического изобретения, которое бы столь широко вторгалось во все сферы человеческой деятельности и преобразовывало их. Однако в этом нет ничего удивительного: повсеместность применения компьютеров вытекает из повсеместности использования информации.

После этого небольшого вступления перейдем к схеме ЭВМ. Ясно, что основу информационной машины должно составлять устройство, обрабатывающее информацию.

Устройство обработки информации — это процессор, главная часть, «мозг» ЭВМ. Помимо обработки информации на него возложена еще одна функция — управление всеми остальными устройствами, организация и координация их взаимодействия.

В ЭВМ первых поколений эти функции выполнялись двумя различными блоками процессора: арифметико-логическим устройством (АЛУ) и устройством управления (УУ). В современных компьютерах АЛУ и УУ часто не разделяются конструктивно, но двойственность функций процессора сохраняется.

Второе главное устройство ЭВМ — память. Она хранит информацию, передает ее на обработку процессору и принимает от него полученную в результате обработки новую информацию.

В соответствии с принципами фон Неймана, на основе которых действуют практически все сегодняшние ЭВМ, программа — это один из видов информации, и хранится она в памяти, как и любая другая информация. Более того, программа и данные принципиально неотличимы, трактовка каждого участка памяти как содержащего программу или данные происходит непо-

средственно во время работы процессора и зависит от хода этой работы. Следовательно, программы можно рассматривать как данные и обрабатывать их, используя ЭВМ (работающую по ранее написанным программам!) для создания новых программ. Этот принцип положен, например, в основу работы трансляторов с языков высокого уровня.

Процессор и память образуют комплекс, способный хранить и перерабатывать информацию, но он является «вещью в себе», ибо не имеет связи с внешним миром. Эту функцию выполняют устройства обмена — ввода и вывода.

Упрощенная концептуальная схема ЭВМ приведена на рисунке.



Рассмотрим некоторые характеристики отдельных частей комплекта.

Процессор. Устройство и принцип работы процессора достаточно подробно рассмотрены в УП, так что расскажем лишь о некоторых важных для пользователя характеристиках процессора.

Первая из них — быстродействие, т. е. скорость, с которой происходит обработка информации. Сегодня, когда производительность даже ПЭВМ приближается к миллиону операций в секунду, этот показатель во многом утратил свою первостепенную важность, но он остается существенным для некоторых приложений, характеризующихся большими объемами вычислений или необходимостью уложиться в жесткие временные рамки. Примеры — инженерные расчеты сложных объектов, прогнозы погоды, управление процессами в реальном времени.

Тем не менее для большинства массовых применений ЭВМ время обработки не является критическим фактором, поэтому самый быстрый компьютер не всегда самый лучший. Не следует забывать и о том, что время решения задачи определяется не только производительностью процессора. Важное значение имеют объем памяти и скорость обмена с внешними устройствами (об этом будет подробнее сказано чуть позже).

34 Еще одна характеристика процессора — система команд. В учебном пособии рассматриваются команды ДВК-2М. Их можно считать типичными: в большинстве машин команды работают с внутренними регистрами процессора, выполняют простейшие операции (словоение, сравнение, пересылка и т. д.), аналогично ДВК реализуются команды перехода, лежащие в основе циклов и ветвлений языков высокого уровня. Тем не менее конкретные наборы команд могут довольно сильно отличаться друг от друга.

Существуют семейства ЭВМ, системы команд которых полностью или частично совпадают. В этом случае говорят о совместимости ЭВМ. Совместимость предполагает возможность переноса программ с одного компьютера на другой, имеющий ту же систему команд.

Массового пользователя система команд интересует именно с точки зрения совместимости: ему необходимо знать, программы с каких компьютеров он сможет перенести на свой и использовать.

Память. Важнейшая характеристика памяти — ее объем. Измеряется память в байтах и кратных им единицах (1К байт=1024 байт, 1М байт=1024К байт).

Ясно, что большая память обеспечивает работу более длинной программы. Кроме того, большой объем памяти позволяет уменьшить количество обменов с внешними устройствами (часто используемые данные можно хранить не на диске или магнитной ленте, а в основной памяти), что увеличивает, иногда довольно существенно, производительность ЭВМ.

Современные супер-ЭВМ имеют память в десятки мегабайт. Персональные

компьютеры, предназначенные для профессиональной работы, — порядка 1М байт. Память массовых персональных ЭВМ, к числу которых относятся и школьные машины, имеет размер 64—256К байт. Меньшая память делает ЭВМ практически недееспособной. Нельзя, например, считать полноценным компьютер БК-0010, имеющий память 32К байта, половина которой уходит на обеспечение изображения на дисплее.

Память ЭВМ делится на оперативную и постоянную (есть еще внешняя память, но о ней мы поговорим чуть позже).

Говоря о памяти, обычно имеют в виду оперативное запоминающее устройство (ОЗУ). В ОЗУ можно занести любую информацию, поступающую от внешних устройств или процессора. После обработки эта информация может быть произвольным образом изменена. При выключении компьютера информация в ОЗУ пропадает.

В постоянное запоминающее устройство (ПЗУ) информация заносится только один раз — при его изготовлении; она сохраняется и при выключении компьютера, ее невозможно изменить. В ПЗУ обычно заносят (на профессиональном жаргоне говорят «зашивают») часто используемые программы, которые нельзя или неудобно загружать в ОЗУ с устройств внешней памяти.

В ПЗУ хранится программа проверки работоспособности и начальной загрузки машины, которая автоматически выполняется при каждом включении ЭВМ. Во многих персональных ЭВМ в ПЗУ хранится интерпретатор Бейсика. В ПЗУ хранятся программы многочисленных специализированных ЭВМ, управляющих процессами и объектами и т. д.

В последнее время получили распространение сменные ПЗУ — небольшие блоки, которые легко можно вставить в специальное гнездо компьютера. Сменные ПЗУ позволяют быстро менять программы, не обращаясь к внешним устройствам. Фактически сменные ПЗУ занимают промежуточное положение между внутренними и внешними устройствами. С точки зрения архитектуры ЭВМ это часть памяти, т. е. внутреннее уст-

ройство, но для пользователя оно, безусловно, выглядит как внешнее.

Внешние устройства ЭВМ. Название «внешние» создает ощущение, что речь идет о чем-то необязательном, не имеющем непосредственного отношения к ЭВМ. К сожалению, это ощущение порой формирует политику производителей вычислительной техники — так появляются на свет некомплектные машины, обманывающие ожидания многочисленных пользователей.

В действительности сегодня именно внешние устройства часто определяют и ограничивают возможные применения ЭВМ. Это легко объяснить: компьютер, когда-то недосыгаемый для пользователя, теперь предельно приблизился к нему, и человек, непосредственно работающий с ЭВМ, хочет, чтобы общение с ней было простым и удобным, так как только тогда он сможет сосредоточиться на содержательной части задачи; а форма общения во многом определяется имеющимся набором внешних устройств.

Современные внешние устройства можно разбить на 2 больших класса: устройства связи с человеком и устройства внешней памяти.

Устройства связи с человеком. Основное устройство, с помощью которого человек общается с современными ЭВМ, — это дисплей, гибрид телевизора и пишущей машинки. Строго говоря; это даже не одно, а два устройства: монитор (он же экран, он же телевизор; под словом «дисплей» тоже часто понимается именно монитор) и клавиатура.

Различные модели клавиатур различаются исполнением клавиш (выделим, например, пленочные клавиатуры), а также их количеством и расположением.

Клавиатура практически не влияет на функциональные возможности ЭВМ, но от нее в значительной мере зависит настроение и работоспособность пользователя. Труднонажимаемые (пленочная клавиатура у БК-0010) или постоянно заедающие («Агат») клавиши способны довести до стрессового состояния даже закаленного программиста. Размещение клавиш — тоже важный эргономический фактор. Недоста-

точное количество клавиш может привести к необходимости использования труднозапоминаемых комбинаций, избыток клавиш делает клавиатуру необозримой. Более подробный анализ должны, видимо, сделать психологи и эргономисты.

А вот о мониторе мы поговорим чуть подробнее. Это основное средство передачи информации от ЭВМ к человеку, и форма представления информации, существенно зависящая от возможностей монитора, имеет первостепенное значение.

Мониторы делятся на алфавитно-цифровые и графические. На алфавитно-цифровом могут изображаться только символы из заранее заданного набора. Наборы символов у разных моделей отличаются незначительно; туда, как правило, входят буквы, цифры, знаки препинания, некоторые специальные знаки. К сожалению, большинство отечественных алфавитно-цифровых мониторов разрешают использовать только прописные русские и латинские буквы, и лишь немногие модели допускают строчные буквы. Некоторые мониторы имеют дополнительные возможности выделения отдельных символов (яркость, подчеркивание и т. д.).

Экран алфавитно-цифрового монитора разбит на строки, а строки — на позиции. В каждую позицию (ее называют еще знакоместом) может быть выведен только один символ. Наиболее распространенный в СССР формат — 24 строки по 80 позиций в каждой. Уменьшение размеров экрана снижает его информационную емкость и делает работу менее удобной.

Алфавитно-цифровые мониторы хороши для обработки текстов, при решении вычислительных задач, но для многих приложений они неудобны или просто неприемлемы. Работа с графиками, чертежами, рисунками требует использования графического монитора, который выводит на экран не отдельные символы, а произвольные изображения; экран делится не на знакоместа, а на точки. Их значительно больше, например: 256×512.

Различают монохромные мониторы с постоянной яркостью (каждая точка на-

ходится в одном из двух состояний: светится или не светится), монохромные с переменной яркостью (яркость точки имеет несколько градаций, изображение приобретает оттенки, приближаясь к нормальному, черно-белому телевизионному) и цветные (каждая точка может иметь любой цвет из заранее заданного набора). Очевидно, что это существенно влияет на функциональные возможности монитора.

Важная характеристика графического монитора — разрешающая способность, определяемая количеством точек на экране. Чем больше точек или чем меньше каждая из них, тем выше качество изображения, менее заметна его точечная структура.

36

У «Ямахи», например, размер экрана 256×192 точки. Этот размер является в некотором роде критическим. Он позволяет создать достаточно сложное и содержательное изображение, но его точечная структура видна слишком явно, реализовать мелкие детали зачастую не удается.

Физически изображение на экране монитора получается по тем же принципам, что и на телевизионном. Оно постоянно регенерируется с высокой частотой, создавая у человека иллюзию устойчивой «картинки».

Проблема здесь в том, что для восстановления изображения его необходимо хранить. Требуется дополнительная память. Ее размеры сравнительно невелики для алфавитно-цифрового монитора (1 байт на символ, около 2К байт для экрана 80×24), несколько больше для графического с постоянной яркостью (1 бит на точку, примерно 7К байт для экрана 300×200 точек), довольно значительны для цветного и монохромного с переменной яркостью (4 бита на точку при 16 цветах (градациях яркости), 1 байт при 256, сотни килобайт для экрана с высокой разрешающей способностью и большим количеством цветов).

На практике часто используются различные хитрости, позволяющие получить изображение при меньшем расходе памяти, но законы информатики неумолимы: сокращая память, мы уменьшаем количество информации на экране мони-

тора, снижая тем самым его изобразительные возможности.

В алфавитно-цифровых мониторах, подключаемым к ЭВМ в качестве терминалов, память обычно встроена непосредственно в монитор. В современных персональных компьютерах видеопамять графического монитора является частью ОЗУ. Таким образом, монитор, оставаясь функционально внешним устройством, становится архитектурно полувнутренним.

В некоторых моделях (например, БК-0010) видеопамять просто выделяется из ОЗУ, при этом программист имеет возможность, изменяя режим построения изображения, регулировать распределение памяти, увеличивая объемы программы и данных за счет снижения информационной емкости экрана и наоборот.

В более сложных конструкциях (например, в «Ямахе») видеопамять добавляется к основному ОЗУ, существуя в виде самостоятельного блока.

Как бы красиво ни было изображение на экране монитора, оно, к сожалению, недолговечно. На практике часто возникает необходимость сохранить полученную от компьютера информацию в удобной для человека форме. Для этого предназначены устройства вывода информации на бумагу: принтеры и графопостроители.

Принтер, или печатающее устройство, предназначен в первую очередь для вывода текстов. В комплект больших ЭВМ обычно входит скоростной барабанный принтер, так называемое АЦПУ (алфавитно-цифровое печатающее устройство), печатающий аналог алфавитно-цифрового монитора. Набор символов в АЦПУ жестко задан (символы выгравированы на специальных барабанах, подобно литерам на рычагах пишущей машинки), печать производится построчно, количество позиций в строке фиксировано.

К персональным компьютерам чаще всего подключают мозаичные принтеры. Они работают не так быстро, как барабанные, но занимают значительно меньше места, а печать получается более качественная. Мозаичный принтер, подобно графическому монитору, «соби-

рает» изображение из отдельных точек. Возможный вариант его устройства показан в книге для учителя. Точечная печать позволяет легко изменять шрифт, выводить на бумагу схемы, графики. Многократное пропечатывание одного места создает эффект оттенков, т. е. дает возможность получать сложные черно-белые рисунки, напоминающие изображение на экране монохроматического монитора с переменной яркостью. Но здесь необходимо отметить одно отличие. Изображение на экране программа может строить в произвольной последовательности, постепенно насыщая его деталями. Принтер печатает только сверху вниз, поэтому рисунок должен быть предварительно полностью сформирован в памяти. Помимо расхода памяти это часто требует очень кропотливого труда программиста.

Этих недостатков нет у другого устройства — графопостроителя. Графопостроитель строит на листе бумаги, уложенном на специальной подставке, изображение. Пишущий узел может свободно перемещаться по листу, поднимать и опускать перо, использовать перья разных цветов. Команды управления графопостроителем по своей сути аналогичны графическим командам учебного пособия. Отметим, что использование графопостроителя не требует дополнительной памяти; изображение может строиться в произвольной последовательности, а перенесенные на бумагу детали сохраняются независимо от того, что происходит в ОЗУ. Таким образом, графопостроитель — незаменимое средство получения сложных графических изображений, обязательный элемент всех систем автоматизированного проектирования. А вот для печати текстов он почти непригоден.

Коротко остановимся на звуковых устройствах ЭВМ. Устройства речевого ввода остаются пока экзотикой, редкостью, а вот звуковой генератор есть практически у всех персональных компьютеров. Это обеспечивает многочисленные музыкальные приложения, звуковое оформление программ и т. д. Из других устройств связи с человеком отметим редкие пока на наших компьютерах «мышки» и джойстики.

«Мышь» — это небольшая коробочка, которую можно катать по столу. Траектория движения передается в ЭВМ и может быть использована, например, для построения графического изображения.

Джойстик — вертикальная ручка на подставке, которую можно отклонить в любом направлении. Направление (а иногда и угол) наклона передается в ЭВМ. Джойстик обычно используется для управления движением объектов на экране компьютера (например, в играх).

Устройства внешней памяти. Внутренняя память ЭВМ имеет два существенных недостатка.

Во-первых, ее объем ограничен. Как бы велик он ни был, туда не войдут одновременно все программы, которые должны работать на машине, а ведь нужны еще и данные.

Во-вторых, информация в ОЗУ пропадает при выключении питания, а ведь хранение информации — одна из важнейших функций ЭВМ.

Ну а в ПЗУ нельзя записать новую информацию.

Для преодоления этих недостатков используются устройства внешней памяти, способные долго хранить большие объемы информации.

Первые устройства внешней памяти использовали бумажные носители информации: перфокарты и перфоленты. Недостатки перфоносителей очевидны: механическая непрочность, невозможность корректировки информации без перезаписи ее на новый носитель, большой расход бумаги, малая скорость доступа к информации. Бумажная память — техника вчерашнего дня.

Основные сегодняшние устройства внешней памяти — накопители на магнитных носителях, лентах и дисках. Существуют уже оптические диски, другие способы хранения информации, но мы не будем на них останавливаться, так как, во-первых, они еще не имеют массового применения, а во-вторых, функционально практически не отличаются от обычных магнитных дисков.

Магнитные устройства внешней памяти могут существенно различаться по размерам и исполнению.

В персональных компьютерах часто

используется бытовой магнитофон с обычной компакт-кассетой. В профессиональных ПЭВМ используется более сложный магнитофон, напоминающий бытовой катушечный. Большие ЭВМ комплектуются скоростными вакуумными накопителями на магнитных лентах.

Разными бывают и диски. На больших машинах работают жесткие диски емкостью 10—100М байт, на персональных компьютерах используют малые гибкие диски объемом 10—100К байт. В последнее время широко распространились винчестерские диски. Они составляют с дисководом одно неразборное целое. Благодаря герметизации дисковода удалось применить некоторые конструкторские ухищрения, повысившие емкость диска до нескольких десятков мегабайт, что компенсирует большие (по сравнению с гибкими дисками) цену и размеры винчестера.

38

Все эти различия носят в основном количественный характер. Качественные же различия между лентами и дисками значительно более существенны, чем между отдельными видами тех и других. Отличия эти вытекают из метода доступа: последовательного для лент и прямого для дисков. Ясно, что прямой доступ, при котором информация может считываться в произвольной последовательности и записываться в произвольное место, значительно быстрее последовательного. Кроме того, он позволяет создавать на дисках сложные файловые структуры и наборы данных, на

ленте практически не реализуемые. Без дисков не могут существовать развитые операционные системы, системы программирования, базы данных. Так что компьютер без диска (например, БК-0010 или ДВК-1) не более чем игрушка.

В последнее время стали использоваться так называемые электронные диски. Отсутствие дисковой памяти (или недостаточную скорость доступа гибких дисков) компенсируют избытком памяти оперативной. В ОЗУ выделяется специальная область, называемая электронным диском. В начале работы этот диск заполняется информацией с ленты или гибкого диска, затем он используется как высокоскоростное устройство с прямым доступом. В конце работы информация должна быть сохранена на реальном устройстве внешней памяти.

Этот прием (использование ОЗУ вместо диска) имеет свое зеркальное отражение — метод виртуальной памяти, когда часть диска становится расширением ОЗУ, компенсируя нехватку основной памяти.

На этом мы закончим краткий обзор устройства ЭВМ. В него не вошло многое; не упомянули мы о машинах нетрадиционной архитектуры, не коснулись очень интересной темы «Компьютер в мире компьютеров» (о сетях, связывающих отдельные ЭВМ в сложные комплексы). Но, увы, нельзя объять необъятное, а мир современных компьютеров практически неисчерпаем.

С. СОЧНЕВ

Изучение алгоритмического языка в курсе информатики

Основная цель курса ОИВТ — научить методам решения задач с помощью ЭВМ, сформировать алгоритмическое мышление у учащихся. Важную роль в ее достижении играет алгоритмический язык.

Правда, схема изучения алгоритмического языка, предложенная в учебном пособии, не до конца последовательна. Кроме того, ряд осложнений создает появление во второй части учебного пособия новых команд алгоритмического языка, изучение которых удобнее проводить в IX классе параллельно с материалом первой части.

Настоящие рекомендации для преподавателей

информатики содержат логически последовательную схему изучения алгоритмического языка и показывают, каким образом можно осуществить переход от изучаемых конструкций алгоритмического языка к языку программирования.

I. Знакомство с алгоритмическим языком. Величины

Алгоритмический язык по своей структуре близок к языкам программирования. Поэтому при знакомстве с ним можно использовать ту же схему, по которой происходит изучение языков прог-

раммирования. Начинать необходимо с алфавита.

Алфавит алгоритмического языка:

1. буквы русского алфавита;
2. 10 цифр;
3. специальные знаки: +, —, х ;, и т. д.

Из символов языка составляются слова, среди которых выделяется особая группа — служебные слова. Служебные слова используются для формирования и описания конструкций языка. Алгоритмический язык содержит два типа конструкций: величины и команды, причем последние делятся на простые и составные. Структура алгоритмического языка приведена на рис. 1.

Изучение конструкций алгоритмического языка более логично начинать не с команд, а с объектов, над которыми эти команды выполняются, — с величин. Заметим, что так обычно поступают и при изучении языков программирования. Величины имеют целый ряд характеристик. Классификацию величин можно провести по четырем признакам (рис. 2). Изучив подробно эти характеристики по предложенной таблице, переходим к знакомству с командами.

II. Команды алгоритмического языка

При знакомстве с командами алгоритмического языка хороший результат дает использование схемы, изображенной на рис. 1. Вначале учащиеся знакомятся с тремя из четырех простых команд, приведенных на схеме. Это команды присваивания

значения: =, arg (которая, по существу, является командой ввода исходных данных), рез (команда вывода). Знание этих команд позволяет учащемуся записать заголовок алгоритма. При этом подчеркивается, что запись заголовка является, собственно говоря, постановкой задачи, так как требует выявления аргументов (что нам дано) и результатов (что надо найти).

Далее переходим к изучению перечисленных в схеме составных команд. Знакомство с первой из них — командой «серия» — приводит нас к понятию линейного алгоритма, две следующие — команды ветвления и выбора — приводят к понятию разветвляющегося алгоритма, а команды повторения и повторения с параметром — к понятию циклического алгоритма. Таким образом, из рассмотрения составных команд логически вытекает необходимость знакомства с типами алгоритмов. Здесь же следует рассмотреть отношения между величинами в качестве условий ветвления или повторения и составные условия и, или, не.

Целесообразно изучение составных команд сопровождать схемами исполнения алгоритмов, содержащих эти команды. При этом полезно использовать методику, предложенную В. Исаковым в статье «Исполнение алгоритма» («Информатика и образование», 1987, № 2). Параллельное изучение команд и их исполнения поможет учащемуся понять, как «работает» та или иная команда, в каких случаях ее надо использовать.

39



Рис. 1. Структура алгоритмического языка ▲

Рис. 2. Величины алгоритмического языка ▼

Величины					
1	постоянные		переменные		
2	числовые			литерные — набор символов <u>лит</u>	графические — геометрические фигуры и линии (граф)
	натуральные <u>нат</u>	целые <u>цел</u>	вещественные <u>вещ</u>		
3	аргументы (<u>arg</u>)		промежуточные	результаты (<u>рез</u>)	
4	простые			табличные — набор пронумерованных величин, имеющих одно имя	

Завершается изучение команд знакомством с простой командой вызова вспомогательного алгоритма. При изучении этой команды следует выделять три этапа в ее выполнении (см. рис. 3). Такой подход позволяет учащимся быстрее усвоить этот материал, а также безболезненно перейти к записи оператора вызова подпрограммы при изучении в дальнейшем языка программирования. Так, в Бейсике, например, первый и третий этапы будут записываться с помощью оператора LET, а второй с помощью GOSUB (см. раздел IV).

Таким образом, знакомство с типами алгоритмов и правилами их записи свелось к изучению конструкций алгоритмического языка, представленных на рис 1. Подробное изучение этой схемы полностью охватывает содержание тем «Алгоритмы. Алгоритмический язык» и «Алгоритмы работы с величинами» программы средней общеобразовательной школы.

III. Алгоритмы решения задач

40

Важную роль для усвоения теоретического материала играет правильный подбор задач. При изучении тем «Алгоритмы. Алгоритмический язык» и «Алгоритмы работы с величинами» достаточно рассмотреть задачи, для которых алгоритмы содержат не более одной составной команды.

Для разветвляющихся процессов это задачи, содержащие либо одну команду ветвления, либо одну команду выбора. При знакомстве с циклическими алгоритмами можно использовать следующие типы задач:

Задача 1 (поиск). Известно, что в таблице A [1:20] есть хотя бы один положительный элемент. Составьте алгоритм нахождения первого положительного элемента таблицы и его порядкового номера.

Задача 2 (табулирование функции). Составьте алгоритм вычисления значений функции $y=x^2$ для целых x от 1 до 100 включительно.

1. Формальным аргументам присвоить их фактические значения

2. Выполнить вспомогательный алгоритм

3. Полученный формальный результат присвоить соответствующей фактической переменной

Задача 3 (построение таблицы). Составьте алгоритм, по которому будут найдены значения элементов таблицы A [1:50], если A [i] определяется по формуле $A [i]=2 \cdot i$

Задача 4 (замена). Составьте алгоритм замены элементов заданной таблицы A [1:20] с нечетными номерами на число 5.

Задача 5 (сумма). Составьте алгоритм вычисления суммы элементов заданной таблицы чисел A [1:10].

Задача 6 (произведение). Составьте алгоритм вычисления произведения элементов заданной таблицы чисел A [1:10].

Алгоритмы для решения этих задач содержат также только одну составную команду. При рассмотрении этих задач следует подчеркнуть, что команду повторения с параметром можно использовать только в том случае, если известны начальное и конечное значения, а также шаг приращения переменной, управляющей циклом. Команду повторения наиболее целесообразно использовать в тех случаях, когда нельзя применить команду повторения с параметром. Тогда для решения задачи 1 следует использовать команду повторения, а для решения задач 2—6 — команду повторения с параметром.

Далее, при изучении темы «Построение алгоритмов для решения задач» рассматриваются уже более сложные задачи по работе с табличными решениями:

Задача 7 (обмен). Даны таблицы A [1:5] и B [1:5]. Составьте алгоритм, по которому элементы таблицы A будут переписаны в таблицу B, а таблицы B — в таблицу A.

Задача 8 (количество). Составьте алгоритм подсчета количества положительных элементов в заданной таблице A [1:30].

Задача 9 (замена с проверкой условия). Составьте алгоритм замены всех отрицательных элементов таблицы A [1:30] на число 25.

Задача 10 (максимум). Составьте алгоритм на-

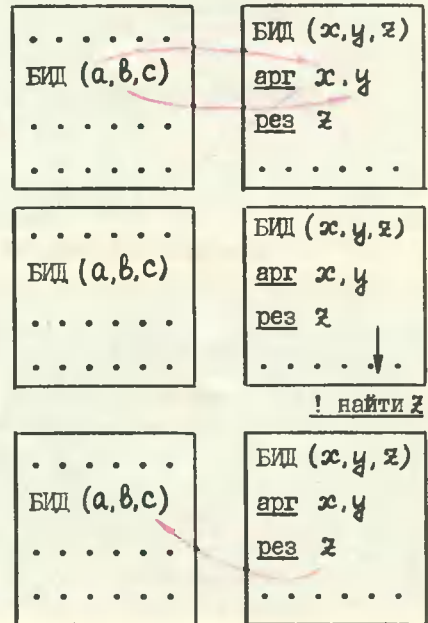


Рис. 3. Команда вызова вспомогательного алгоритма

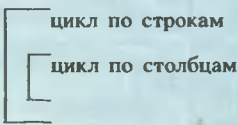
сумма произведение	$S:=0$ $P:=1$	$S:=S+A[i]$ $P:=P \cdot A[i]$
обмен	A B	$C:=B[i]$ $B[i]:=A[i]$ $A[i]:=C$
количество	$K:=0$	если $A[i]$ удовлетворяет условию, то $K:=K+1$
максимум	$M:=A[i]$	если $A[i] > M$, то $M:=A[i]$
прямоугольная таблица		

Рис. 4. Алгоритмы работы с табличными величинами

хождения максимального элемента заданной таблицы чисел A [1:100].

Задача 11 (прямоугольная таблица). Составьте алгоритм построения таблицы чисел B [1:3, 1:5], если $B[i, j]=i+j$.

Только первая из этих задач решается с помощью одной составной команды. Для решения остальных необходимо использовать две составные команды: для задач 8—10 это команды повторения с параметром и ветвления, а для задачи 11 — две команды повторения с параметром. Таким образом, при изучении темы «Построение алгоритмов для решения задач» осуществляется постепенный переход от более простых задач к более сложным.

Наиболее существенные моменты решения этих задач можно изобразить в виде таблицы (см. рис. 4). С помощью такой таблицы можно провести закрепление и обобщение знаний учащихся по алгоритмам работы с табличными величинами. Большинство решаемых учащимися задач можно свести к задачам, приведенным в таблице. Например, алгоритм «упорядочение» (см. с. 60 части 1 учебного пособия) сводится к алгоритмам «максимум» («минимум») и «обмен» и т. д. В то же время у учащихся не должно создаваться впечатление, что с помощью такой таблицы можно решить все задачи без исключения. Для этого на уроках следует рассмотреть и более сложные задачи (например, ветвление внутри двух циклов) и оригинальные задачи, решение которые требует нестандартного подхода (например, задачи из упражнений для повторения, часть 2 учебного пособия, с. 100—103).

Полученные знания учащиеся должны использовать для построения алгоритмов решения задач из курсов физики и математики (схема Горнера, вычисление площади криволинейной трапеции и т. д.). Изучение темы «Построение алгоритмов для решения задач» завершается знакомством с графическими и литерными величинами и командами

алгоритмического языка, используемыми для работы с ними.

IV. Переход к языку программирования

Опираясь на знания учащихся, полученные при изучении алгоритмического языка, несложно ознакомить учащихся с особенностями того или иного языка программирования. Покажем, как осуществляется такой переход от алгоритмического языка к языку программирования на примере Бейсика. Заметим, что Бейсик имеет существенные отличия от алгоритмического языка, тем не менее при изучении Бейсика удается провести полную аналогию между этими двумя языками.

Изучение языка Бейсик (так же как и алгоритмического) начинается с алфавита. Алфавит Бейсика включает в себя:

- 26 заглавных латинских букв;
 - 10 цифр;
 - Специальные знаки: + — / : , . и т. д.
- Далее переходим к рассмотрению структуры языка. Бейсик, как и алгоритмический язык, включает в себя два типа конструкций:
- команды (операторы);
 - величины.

В отличие от алгоритмического языка команды в Бейсике не делятся на простые и составные. Изучение величин в Бейсике идет практически по той же схеме, что и в алгоритмическом языке (см. рис. 2). В зависимости от версии Бейсика в эту схему вносятся незначительные изменения. Каждой конструкции алгоритмического языка ставится в соответствие оператор (или группа операторов) Бейсика (см. рис. 5). При записи таблицы, представленной на рис. 5, использовалась версия Бейсика для микро-ЭВМ «Электроника ДЗ-28», в которой в операторе IF...THEN после слова THEN можно указать не только номер строки, но и любую последовательность операторов Бейсика. Для других версий в эту таблицу также надо внести некото-

<u>алг</u> <u>кон</u>	REM END
<u>арг</u> <u>рез</u> присвоить значение A:=5	INPUT PRINT LET LET A=5 (A=5)
<u>если</u> условие <u>то</u> серия 1 <u>иначе</u> серия 2 <u>все</u>	10 IF условие THEN серия 1:GOTO 30 20 серия 2 30
<u>выбор</u> <u>при</u> условие 1: серия 1 <u>при</u> условие 2: серия 2 <u>при</u> условие 3: серия 3 <u>иначе</u> серия <u>все</u>	10 IF условие 1 THEN серия 1:GOTO 50 20 IF условие 2 THEN серия 2:GOTO 50 30 IF условие 3 THEN серия 3:GOTO 50 40 серия 50
<u>пока</u> $x \neq 0$ (<u>пока не</u> $x=0$) <u>нц</u> серия <u>кц</u>	10 IF X=0 THEN 70 20 } .. } серия 50 } 60 GOTO 10 70
<u>для</u> i <u>от</u> i_n <u>до</u> i_k <u>шаг</u> Δi <u>нц</u> серия <u>кц</u>	10 FOR I=IN TO IK STEP DI 20 } .. } серия 50 } 60 NEXT I
команда вызова вспомогательного алгоритма (ср. с рис. 3) 1 этап 2 этап 3 этап	40 X=A:Y=B 50 GOSUB ... 60 C=Z

Рис. 5. Основные конструкции языка Бейсик

рые изменения, которые не носят принципиального характера.

Используя такую таблицу, учащиеся могут в буквальном смысле «переводить» алгоритмы с алгоритмического языка на Бейсик.

Следуя предложенной методике, удастся пол-

ностью провести алгоритмическую линию в курсе «Основы информатики и вычислительной техники». Алгоритмический язык выступает в этом случае действительно необходимым звеном при подготовке учащихся к работе на ЭВМ, предворяет изучение одного из языков программирования.

Л. КОНОВАЛОВ
г. Ачинск

Из опыта преподавания темы: «Табличные величины. Алгоритмы работы с табличными величинами»

В методическом пособии для учителей дается понятие табличной величины как сложно организованной, или структурной. После рассмотрения различных примеров линейных таблиц и записи их при помощи служебных слов полезно показать учащимся структурную запись табличной величины, которая дает более полное представление о величине и с успехом может быть

использована при объяснении алгоритмов работы с табличными величинами.

Пусть дана линейная таблица цел таб В [1:5]

Ее структурная запись состоит из двух строк: первая строка — порядковые номера элементов таблицы (счет которых можно «поручить» любой промежуточной переменной, например, i);

вторая — значения элементов таблицы V [1], V [2], ..., V [5], общий вид которых — $V[i]$

i	1	2	3	4	5
V	$V[1]$	$V[2]$	$V[3]$	$V[4]$	$V[5]$
Пример V	9	3	5	7	6

Отсюда легко увидеть порядковые номера и количество элементов таблицы, проще объяснить введение промежуточной переменной i , имени табличной величины V . По значениям элементов определяется тип величины.

Задач обработки информации, заданной в форме таблиц, множество. Укажем типовые.

1. Задачи на изменение значений элементов таблицы:

- возвести значения элементов таблицы в квадрат, куб, n -ю степень;
- извлечь корень из каждого элемента;
- увеличить (уменьшить) значения элементов таблицы на m единиц или в n раз;
- заполнить таблицу другими значениями элементов.

2. Задачи на поиск элемента таблицы с заданным свойством:

- определить количество элементов таблицы, значения которых равны M ;
- найти максимальный (минимальный) элемент таблицы;
- определить первый номер элемента, значение которого равно P .

3. Задачи на нахождение суммы значений элементов таблицы (всех элементов; четных или нечетных; равных M и т. д.).

4. Упорядочение элементов таблицы по какому-либо признаку (например, по возрастанию значений элементов).

Рассмотрим примеры построения алгоритмов работы с табличными величинами с использованием структурной записи.

Учащиеся хорошо понимают и усваивают построение алгоритма, если делать анализ по условию задачи и отдельно по математической модели решения задачи. Первый анализ необходим для написания заголовка алгоритма.

Пример 1. Написать алгоритм, увеличивающий значения элементов цел таб V [1:n] на A .

Анализ 1. Запишем указанную таблицу структурно. Из обеих записей видно, что элементы таблицы $V[1]$, $V[2]$, ... $V[n]$ — целые числа, и их имя V . Количество элементов в таблице — n , n — целое число. Аргументами алгоритма являются n и V . Результатом будет новая таблица V с тем же количеством элементов, значения которых увеличены на A . Название алгоритма — увеличение на A . Следовательно, заголовок алгоритма таков:

алг увеличение на A (цел n , цел таб V [1:n])
арг n , V
рез V

Анализ 2. Введем промежуточную переменную i для счета элементов таблицы. Начнем просмотр с первого элемента, поэтому вначале $i:=1$.

Берем первый элемент старой таблицы $V[1]$ и увеличиваем его на A , получим первый элемент новой таблицы, что запишется: $V[1]:=V[1]+A$

Затем переходим ко второму элементу старой таблицы командой $i:=i+1$

Берем второй элемент старой таблицы $V[2]$ и увеличиваем его на A , получим второй элемент новой таблицы.

$V[2]:=V[2]+A$

Переходим к 3-му элементу таблицы командой $i:=i+1$

и так далее, пока не переберем все элементы таблицы, а их n .

При увеличении каждого элемента таблицы на A выполняются только две команды, которые в общем виде запишутся так:

$V[i]:=V[i]+A$

$i:=i+1$

Сколько элементов в таблице, столько должно быть и повторений этих команд.

Условием цикла является $i \leq n$.

Следовательно, тело алгоритма запишется так:

нач цел i
 $i:=1$
пока $i \leq n$
нц $V[i]:=V[i]+A$
 $i:=i+1$

кц

кон

Пример 2. Построить алгоритм нахождения максимального элемента таблицы цел таб V [1:n].

Анализ 1. Анализ аналогичен примеру 1. Результат задачи, его имя и тип определяется из условия задачи. Результатом данной задачи является максимальный элемент с именем X , его тип — целый. Название алгоритма — макс-элемент. Заголовок алгоритма: алг максэлемент (цел n , цел таб V [1:n], цел X)

арг n , V
рез X

Анализ 2. Запишем таблицу структурно. Возьмем первый элемент, $i:=1$. Будем считать, что максимальный элемент равен значению первого элемента таблицы, т. е. $X:=V[1]$. Затем начнем сравнивать значение первого элемента со 2-м, 3-м и т. д. При сравнении возможны два случая:

- 1) если $X \geq V[2]$, то перейдем к 3-му элементу командой $i:=i+1$;
- 2) если $X < V[2]$, то $X:=V[2]$ и перейдем к 3-му элементу.

При сравнении команды повторяются, следовательно, алгоритм циклический с условием $i \leq n$. Команды тела цикла в общем виде запишутся так:

если $X < V[i]$
то $X:=V[i]$
все

$i:=i+1$

Тело алгоритма:

нач цел i
 $i:=1$
пока $i \leq n$
нц
если $X < V[i]$
то $X:=V[i]$
все
 $i:=i+1$

кц

кон

Рассмотренные схемы анализа задач по обработке табличных величин легко усваиваются учащимися. В заключение темы можно дать домашнюю контрольную работу.

Программа факультативного курса «Избранные главы информатики»

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Предлагаемый факультативный курс имеет своей целью углубить знания учащихся в области наиболее актуальных направлений информатики, приобретенных ими при прохождении основного курса «Основ информатики и вычислительной техники» и обеспечить органичный переход к профессиональному изучению и использованию информатики в своей будущей деятельности.

Задачи факультативного курса:

сформировать у учащихся знания об основных компонентах системного программного обеспечения и машинной графики, информационных систем, баз данных, баз знаний и их роли в системах искусственного интеллекта;

44

привить навыки формализованного описания поставленных задач, чтобы обеспечить возможность использовать для их решения развитое программное обеспечение;

показать роль графической формы представления информации в различных областях человеческой деятельности и раскрыть основные принципы построения систем, использующих машинную графику;

обеспечить углубленное понимание основных принципов работы компонентов современного программного обеспечения, таких, как редакторы текстов, графические редакторы, компиляторы, экспертные системы и др.;

рассказать о различных способах представления информации в ЭВМ и показать, какое влияние оказывает выбор способа на эффективность применяемых алгоритмов;

сформировать необходимые навыки разработки простейших программ, быстрого освоения и грамотного использования готового системного и прикладного программного обеспечения.

Программа факультатива согласована по своему содержанию с программой основного курса «Основы информатики и вычислительной техники» для IX—X классов. Учитель может по своему усмотрению выбрать две темы курса. Занятия проводятся из расчета 1 час в неделю: 34 часа в IX классе и 34 часа в X классе.

Можно рекомендовать следующее построение курса. В IX классе — машинная графика, в X классе — системное программирование. Другой вариант: в IX классе — базы данных и информационные системы, в X классе — представление знаний и искусственный интеллект. Хотя каждая из тем независима от другой, предложенные варианты пронизаны внутренним единством и, в сущности, образуют комплексы знаний об интегрированных системах готового программного обеспечения. Примерами систем, отвечающих первому варианту, являются автоматизированные рабочие места (АРМ) специалистов разного профиля, для второго варианта — это интеллектуальные системы управления базами данных (СУБД) и экспертные системы.

Тема «Машинная графика» содержит материал, углубляющий знания, получаемые учащимися при прохождении основного курса. В связи с тем, что содержание темы «Машинная графика» вызывает наибольший интерес у учащихся, изложение построено таким образом, чтобы на большом количестве примеров раскрыть практические методы обработки графической информации. Вместе с тем определенное внимание уделяется систематическому изложению тех теоретических положений, которые не зависят от архитектуры используемой ЭВМ и составляют основу новой современной науки — интерактивной машинной графики. Это относится к таким разделам, как структурирование графической информации, геометрические преобразования, оживление изображений. Естественно, авторы не могли обойти вопроса построения диалоговых систем на основе методов машинной графики, так как они являются основой построения современных автоматизированных рабочих мест. Предполагается, что составной частью этого курса является графический пакет, включающий необходимые алгоритмы работы с графическими величинами. Реализация отдельных алгоритмов этого пакета в конкретной системе программирования является содержанием практических работ.

Тема «Системное программирование» содержит описание базовых, ставших уже фундаментальными компонентов программного обеспечения (ПО) ЭВМ. Это так называемое системное программное обеспечение. Оно поддерживает необходимые условия для эффективной разработки и функционирования прикладных программ, создаваемых пользователями ЭВМ. При этом учитываются как потребности пользователей, так и эффективное использование аппаратных ресурсов ЭВМ. Тема включает «Введение», в котором выделена особая роль системного программиста в индустрии обработки данных в настоящем и будущем. Включены традиционные разделы о системах подготовки текста, системах трансляции и операционных системах. Большой объем материала потребовал выделения ключевых понятий по каждому из разделов. Это структуры данных для представления текста в памяти ЭВМ, понятия о формальных грамматиках и грамматическом разборе, понятие об операционной системе как о средстве управления ресурсами ЭВМ. Нетрадиционным является раздел о современных интегрированных системах ПО персональных ЭВМ. Он позволяет связать воедино знания и навыки, полученные при изучении предыдущих разделов, и очертить тенденции будущего развития системного программирования. Предусмотренные в программе по этой теме практические работы не только способствуют углублению понимания курса и приобретению практических навыков работы с системным программным обеспечением, но и могут внести реальный вклад в расширение системного ПО школьного КУВТ. При изучении темы учителю следует обратить внимание

на наиболее сложный раздел — операционные системы (ОС). Его изучение должно сформировать у учащихся понимание основных принципов работы ОС.

Тема «Базы данных и информационные системы» посвящена двум вопросам. Первый связан с изучением способов организации, хранения и доступа к данным большого объема, при котором выполнение основных операций — поиск и сортировка — оценивается по затратам памяти и времени. Этому вопросу уделяется сравнительно мало внимания в учебной литературе. Исключением является известная монография Н. Вирта «Алгоритмы + структуры данных = программы», однако она рассчитана на подготовленных программистов. Второй вопрос связан с изучением информационных систем и баз данных. Изложение построено таким образом, чтобы постепенно подвести читателя к основной идее — независимости формы представления данных от прикладной программы и к тому, как эта идея реализуется в современных системах управления базами данных (СУБД). Изучаются различные способы организации данных, начиная с простейшего, основанного на последовательном доступе и кончая табличным представлением, используемым при организации реляционных баз данных. В заключение даются первоначальные сведения о СУБД, как о специальном программном средстве, предназначенном для описания, манипулирования и доступа к данным. Для проведения практических работ предполагается использовать как специально разработанное программное обеспечение, позволяющее изучать различные структуры данных и операции над ними, так и готовое программное обеспечение — учебную базу данных, с помощью которой учащиеся смогут решать конкретные задачи в различных предметных областях.

Тема «Представление знаний и искусственный интеллект» содержит описание новых, активно развивающихся компонентов программного обеспечения ЭВМ. В наше время еще только происходит переход от исследований в этой области к промышленному внедрению систем искусственного интеллекта (ИИ). Прежде всего, это так называемые «расчетно-логические» и экспертные системы (ЭС), а также разработка специальных компьютеров для систем ИИ — ЭВМ пятого поколения. Исследования и разработки в этой области вбирают в себя достижения всех остальных разделов информатики, в том числе тех, которые изучаются в этом курсе. Стратегическая цель разработок — превращение компьютера в активного помощника и советчика каждого специалиста (непрограммиста) с постепенной передачей ЭВМ функций по решению задач, традиционно считавшихся интеллектуальными.

Во введении рассмотрены пути развития систем ИИ. Следующие два раздела — планирование действий и описание знаний — взаимосвязаны. В них рассмотрены основные компоненты систем ИИ. Следует подчеркнуть здесь основную идею о том, что поиск решения интеллектуальных задач, как правило, приводит к перебору астрономического числа альтернативных вариантов, который невыполним ни на каких ЭВМ. В этом случае расширение и особая организация базы знаний об этой задаче позволяют существенно сократить перебор или совсем избежать его. В следующем разделе

рассмотрены некоторые реализации систем ИИ. Более глубоко изучаются ЭС. Причиной этого послужило начало их проникновения во все области человеческой деятельности: от постановки медицинских диагнозов до оптимизации системы управления космическим кораблем. В заключительном разделе рассмотрены перспективы систем ИИ. Прежде всего, это проекты и первые результаты разработки компьютеров пятого поколения. Предусмотренные программой по этой теме практические работы предполагают наличие в школьном КУВТ довольно богатого ПО: программы — конструктора игр, интерпретаторов языков Лисп и Пролог, модельной ЭС. Использование этого ПО должно превратить изучение наиболее сложной темы курса в увлекательное «конструирование».

II. СОДЕРЖАНИЕ ОБУЧЕНИЯ

Машинная графика (34 ч)

Введение (2 ч)

Предмет изучения машинной графики. Области применения машинной графики. Аппаратные средства машинной графики. Состав программного обеспечения.

Практические работы

1. Демонстрация функционирования интерактивной вычислительной системы (графической).
2. Демонстрация робототехнической модели (на примере готовой программы).

Первоначальные сведения о машинной графике (6 ч)

Экран дисплея с точки зрения программиста. Понятие о степени разрешения экрана. Простейшие элементы изображения — точка и знак. Примеры получения графических изображений с помощью знаков.

Видеопамять и ее назначение. Структура видеопамати для экранов различной степени разрешения.

Цвет в машинной графике. Программирование цветных изображений на экранах различных уровней разрешения.

Вывод плоских примитивов (отрезков, прямоугольников, окружностей) и кривых.

Построение графиков функциональных зависимостей. Вывод текстовой информации на экраны высокого разрешения.

Практические работы

1. Построение графических изображений с использованием примитивов.
2. Работа с готовым пакетом программ построения графиков функциональных зависимостей.

Двумерная машинная графика: методы хранения и преобразования плоских изображений (15 ч)

Мировые координаты и координаты экрана. Понятие окна и поля вывода. Видовая операция и операция отсечения.

Сегментация изображений. База данных для хранения графической информации и ее структура. Состав операций обслуживания графической базы данных.

Геометрические преобразования на плоскости:

поворот, перенос, масштабирование и зеркальное отображение. Реализация геометрических преобразований над графическими объектами.

Оживление плоских изображений. Основные методы имитации движения графических объектов и их реализация на ЭВМ.

Практические работы

1. Разработка учебных баз данных для хранения графической информации.
2. Программирование основных геометрических преобразований.
3. Реализация простейших методов оживления плоских изображений.

Диалоговые методы обработки графической информации (10 ч)

*Различие между пакетным и диалоговым режимом обработки информации на ЭВМ. Состав программных средств обеспечения диалогового режима.

Реализация ввода графической информации и основные логические устройства ввода (селектор, валкуатор, локатор).

Графическое и текстовое меню. Подсказки. Добавление, удаление, идентификация и детализация изображения графических объектов.

Практическая работа

1. Разработка учебного графического редактора.
2. Реализация отдельных программ, выполняющих различные функции диалоговых графических систем.

Литература

1. Ньюмен У., Спрулл Р. Основы интерактивной машинной графики. М., 1976.
2. Роджерс Д., Адамс А. Математические основы машинной графики. М., 1981.
3. Гилрой В. Интерактивная машинная графика. М., 1981.
4. Баяковский Ю. М., Галактионов В. А., Михайлова Т. Н. Графор. Графическое расширение Фортрана. М., 1985.
5. Михайленко В. Е., Анпилогова А. А. Справочник по машинной графике. К., 1984.
6. Фоли Дж., Вэн Дэм А. Основы интерактивной машинной графики, тт. 1, 2. М., 1985.

Системное программирование (34 ч)

Введение (2 ч)

Системное и прикладное программное обеспечение. Состав системного программного обеспечения. Задачи системного программирования. Роль системного программирования в системе обработки данных.

Практические работы

1. Демонстрация работы с редактором и компилятором.
2. Диалог с операционной системой.

Системы подготовки текста (7 ч)

Редакторы текста: пакетные и экранные. Форматоры текста. Структуры данных для представления текста в оперативной и внешней памяти ЭВМ. Отображение текста на экране дисплея.

Система команд и структура учебного редактора текста.

Практические работы

1. Освоение основных команд учебного редактора текста. Ознакомление с функциями его основных подпрограмм.
2. Разработка одной-двух дополнительных команд учебного редактора, используя его подпрограммы в качестве конструктора.

Системы трансляции (9 ч)

Многообразие и классы языков программирования. Машинные языки. Компиляторы и интерпретаторы. Понятие о грамматике языка. Грамматический разбор. Принципиальная схема трансляции. Оптимизация объектного кода. Редакторы связей и загрузчики. Структура учебного интерпретатора алгоритмического языка.

Практические работы

1. Освоение работы с компилятором на примерах простейших программ.
2. Разработка расширений учебного интерпретатора, позволяющих интерпретировать некоторые минимальные расширения алгоритмического языка.

Операционные системы (10 ч)

ОС как средство управления абсолютными (аппаратными) ресурсами ЭВМ на основе предоставления пользователю относительных ресурсов. Виды ресурсов ЭВМ. Управление процессором, памятью, внешними устройствами. Мультипрограммный режим. Виртуальная память. Режим разделения времени. Системы реального времени. Командные языки. Примеры ОС.

Практические работы

1. Ознакомление на практике с командным языком диалоговой ОС.
2. Разработка простейших программ на командном языке ОС.

Интегрированные системы программного обеспечения персональных ЭВМ (6 ч)

Инструментальные системы программирования. Системы обработки текстов. Объектно-ориентированный подход. Редакторы-интерпретаторы. Персональные системы управления базами данных. Крупноблочные электронные таблицы. Системы деловой графики. Автоматизированные рабочие места.

Практические работы

1. Ознакомительная работа с одной из систем интегрированного программного обеспечения.
2. Разработка простейшей программы с использованием инструментальной системы программирования.

Литература для учащихся

1. Современный компьютер. Сборник научно-популярных статей. Пер. с англ. М., 1986.
2. Громов Г. Р. Национальные информационные ресурсы. Проблемы промышленной эксплуатации. М., 1985.

Литература для учителя

1. Бахарев И. А., Горлин А. И. Операционные системы / серия «Математика, кибернетика». М., 1986.

2. Вирт Н. Алгоритмы+структуры данных== программы. М., 1985.

3. Бауэр Ф. Л., Гооз Г. Информатика. Вводный курс. М., 1976.

4. Трахтенгерц Э. А. Как работают операционные системы. М., 1978.

5. Баурн С. Операционная система UNIX. М., 1986.

6. Брябин В. М. Профессиональные персональные ЭВМ / в сб. Прикладная информатика. Вып. 2/7 / М., 1984.

7. Уэйт М., Ангермейер Дж. Операционная система CP/M. М., 1986.

8. Жук Д. М., Мартынюк В. А. Технические средства и операционные системы САПР (в 9-ти книгах). Книга 2. М., 1986.

9. Компьютеры. Справочное руководство в 3-х томах (под редакцией Г. Хелмса). Пер. с англ. М., 1986.

Базы данных и информационные системы (34 ч)

Введение (2 ч)

Понятие о данных, как о сведениях, отражающих состояние реального мира. Интуитивное представление об информационной системе, как о средствах ввода, хранения и выдачи данных по запросу. Примеры реальных информационных систем и их роль в жизни человека: библиографическое обслуживание, бухгалтерский учет, автоматизированные системы управления.

Практическая работа

1. Демонстрация функционирования учебной информационной системы «Картотека книжного фонда».

Методы хранения, доступа и обработки информации при работе с простейшей информационной системой (13 ч)

Первоначальное представление об организации хранения и доступа к данным. Структурные элементы данных: поле, логическая запись, картотека (файл).

Методы доступа к данным: последовательный, библиотечный, прямой.

Выполнение операций над картотеккой: формирование, исключение, поиск, сортировка.

Простейшая база данных как совокупность картотек. Методы устранения дублирования информации. Понятие инвертированного файла.

Практические работы

1. Реализация методов хранения и доступа к данным на одном из языков программирования (Бейсик, Паскаль, Рапира).

2. Работа с учебной программой «Изучение типовых структур данных».

Структуры данных и способы их реализации.

Реляционная база данных (12 ч)

Внешние требования к информационным системам: время получения ответа на запрос, отсутствие дублирования данных, надежность хранения данных.

Структура данных, как отражение связей между элементами данных. Уровни представления данных: концептуальный, логический, физический.

Последовательная структура данных: связанный список. Основные операции над связанными списками.

Иерархическая структура данных: дерево. Поиск информации в древовидной структуре. Понятие о сбалансированном дереве.

Табличное представление данных. Основные понятия, связанные с реляционной базой данных: домены, атрибуты, кортежи, отношения. Примеры выполнения операций над отношениями.

Практические работы

1. Реализация последовательной и древовидной структуры в языках программирования Паскаль, Рапира.

2. Работа с учебной программой «Изучение типовых структур данных» / разделы «Связанный список», «Дерево»/.

Системы управления базами данных (7 ч)

Требования независимости описания данных от логики программы. СУБД как программное средство описания, манипулирования и доступа к данным. Основные требования к СУБД. Анализ взаимодействия программы пользователя и операционной системы компьютера с СУБД.

Практические работы

1. Работа с учебной базой данных: ввод информации, обновление информации, поиск по ключу.

2. Решение задач, связанных с использованием баз данных в различных предметных областях: — составление справочных данных по физике, географии, истории, биологии; — составление словарей иностранных слов и терминов;

— кадровый учет;

— составление и обновление библиографии книжного фонда.

Литература

1. Тиори Т., Фрай Дж. Проектирование структур данных / т. 1, 2 / М., 1985.

2. Мартин Дж. Организация баз данных в вычислительных системах. М., 1978.

3. Крилицкий Н. А., Миронов Г. А., Фролов Г. Д. Автоматизированные информационные системы / Под ред. А. А. Дородницына. М., 1982.

4. Мелдман М. RISS — система управления реляционными базами данных для малых ЭВМ. М., 1982.

Представление знаний и искусственный интеллект (34 ч)

Введение (2 ч)

От данных к знаниям. Два пути развития: укрупнение команд языка и выделение данных в самостоятельные объекты. Преобразование данных в знания. Особенности знаний. Модели описания знаний и системы планирования действий.

Практические работы

1. Работа с учебной базой данных. Выявление ее достоинств и недостатков.

2. Работа со школьной экспертной системой в режиме ознакомления.

Системы планирования действий (8 ч)

Перебор альтернативных действий в системах планирования. Решение переборных задач. Методы сокращения перебора. Сокращение перебора в

игровых задачах. Роль знаний. Ознакомление с конструктором игр — диалоговой системой для синтеза игровых программ.

Практические работы

1. Изучение на практике конструктора игр. Апробация демонстрационных игр.

2. Решение задачи сокращения перебора в процессе разработки (с помощью конструктора) новой игровой программы.

Модели описаний (10 ч)

Процедурные модели. Системы продукций. Необходимость явного описания алгоритма в языках программирования, основанных на системе продукций. Примеры. Особая роль языка Лисп. Декларативные модели, как системы, не требующие описания алгоритма. Примеры. Понятие о языке Пролог. Базы знаний.

Практические работы

1. Демонстрация программ синтеза новых программ на языке Лисп.

2. Диалог с программой на языке Пролог.

Прикладные системы искусственного интеллекта (10 ч)

Интеллектуальные системы автоматизации программирования. Расчетно-логические системы. Экспертные системы (ЭС). Режимы работы пользователя с ЭС. Структура типовой ЭС. Средства программирования ЭС. Работа с экспертами. Примеры ЭС.

Практические работы

1. Работа с модельной школьной ЭС в режиме «ученика».

2. Работа с модельной школьной ЭС в режиме «учителя».

Перспективы систем искусственного интеллекта (3 ч)

Пятое поколение ЭВМ (ВС-5) — компьютеры для систем искусственного интеллекта. Общая схема ВС-5. Подсистемы диалога, планирования и ведения базы знаний. Аппаратная поддержка системы.

Практические работы

1. Подготовка и обсуждение рефератов по достигнутому в мире результатам в разработке ВС-5.

Литература для учащихся

1. Современный компьютер. Сб. научно-популярных статей. Пер. с англ. М., 1986.

2. Нивергельт Ю., Фаррар Дж., Рейнгольд Э. Машинный подход к решению математических задач. М., 1977.

3. Кибернетика. Становление информатики. М., 1986.

Литература для учителя

1. Уинстон П. Искусственный интеллект. М., 1980.

2. Эндрю М. Искусственный интеллект. М., 1985.

3. Тьугу Э. Х. Концептуальное программирование. М., 1986.

4. Нильсон Н. Искусственный интеллект. Подход на основе эвристического программирования. М., 1972.

48

Т. СЕРГЕЕВА, А. ЧЕРНЯВСКАЯ

Дидактические требования к компьютерным обучающим программам

Разработка дидактических требований к компьютерным обучающим программам (КОП)* является сегодня одной из наиболее актуальных проблем компьютеризации обучения. От успешности ее решения зависит качество создаваемых программ, а следовательно, и эффективность их использования в учебном процессе.

В настоящее время существует несколько подходов к разработке таких требований. На начальных этапах компьютеризации учебного процесса требования к КОП представляли собой результат обобщения опыта создания и использования компьютерных обучающих программ. Однако простое воспроизведение существующего опыта, сыграв свою положительную роль, оказывается недостаточным, поскольку эмпирический подход не

позволяет в полной мере отразить существенные, наиболее устойчивые связи процесса обучения, его закономерности.

Ряд исследователей [см., например, 1, 6], справедливо критикуя такой подход, предлагают при разработке дидактических требований к КОП исходить из содержания дидактических принципов. Правомерность подобного подхода не вызывает сомнений, так как положение о том, что дидактические требования всегда основываются на дидактических принципах, является в дидактике общепризнанным, однако пути его реализации не всегда представляются обоснованными. Сегодня уже сделаны попытки выделить дидактические требования к КОП исходя из традиционного содержания принципов обучения, что, на наш взгляд, не позволило в полной мере придать этим требованиям достаточно конструктивный характер. Другие исследователи считают необходимым создание *особой системы* принципов обучения, отражающей специфику учебного процесса в условиях приме-

* Определение КОП см.: Мирская А., Сергеева Т. Обучающие программы оценивает практика // Информатика и образование. 1987. № 6.

нения ЭВМ. Подобная позиция основана на убеждении, что использование ЭВМ в обучении ведет к коренной перестройке последнего, делая его качественно отличным от традиционного.

Однако если исходить из понимания дидактических принципов как *общих* положений или руководств к действию, адекватных любой педагогической ситуации, то попытки подобного пересмотра следует признать малоубедительными. ЭВМ — лишь *средство* осуществления деятельности обучения, логика функционирования которого определяется теми же дидактическими задачами, которые ставятся и при использовании более традиционных технических средств, при безмашинном обучении. ЭВМ наряду с другими средствами обучения, а также методами и формами обучения является одним из способов реализации дидактических принципов, следовательно, последние должны задавать направление в решении вопросов использования компьютеров, а не наоборот.

Таким образом, ни в какой мере не отрицая специфику компьютерного обучения, мы полагаем, что оно подчиняется той же системе дидактических принципов, что и безмашинное обучение, но при условии, что система таких принципов и содержание каждого из них оптимизировано на основе современных данных психологической и педагогической наук. Следовательно, речь должна идти не о замене традиционных дидактических принципов, а о пересмотре и наполнении таким содержанием, которое позволило бы конструктивно использовать их в любых ситуациях обучения, в частности организованных с помощью ЭВМ. В связи с этим требованием инструментальности, как критерий отнесения тех или иных положений к категории «принцип дидактики», должно быть конкретизировано указанием на возможность его применения и в обучении с использованием компьютеров. Только в этом случае тот или иной дидактический принцип будет действительно носить конструктивный характер.

Анализ литературы [1—4; 5, 7, 9 и др.] показал, что некоторые принципы обучения не удовлетворяют современным требованиям и нуждаются в коренном пересмотре. Кроме того, в дидактике пока отсутствует единое мнение в определении системы этих принципов.

В последние годы на основе использования современных теоретических достижений в области обучения перестраивается система традиционных дидактических принципов. Наиболее продуктивным, по нашему мнению, представляется пересмотр принципов обучения с позиции деятельностного подхода к процессу обучения. Так, конкретизировано и представлено на инструментальном уровне содержание принципа наглядности [3, 5, 7], доступности [1], систематичности и последовательности [9], принципа сознательности [4]. В дидактике также оптимизировано содержание таких принципов обучения, как принцип индивидуализации обучения [2], принцип активности [6] и др.

При выявлении дидактических требований к КОП мы прежде всего ориентировались на те принципы обучения, содержание которых пересмотрено на основе современных теоретических достижений в области педагогики и психологии, что и позволило использовать их в качестве

основы для исследования данной дидактической проблемы.

Одним из ведущих дидактических принципов является принцип *научности*. Согласно этому принципу, в содержание обучения должны включаться не только устоявшиеся в науке знания, но и наиболее фундаментальные проблемы современной науки и перспективы ее развития. В соответствии с сегодняшними представлениями, принцип научности определяет не только отбор содержания учебного материала, но и способы его усвоения, адекватные современному научному знанию. В связи с этим предполагается формирование у учащихся умений и навыков научного поиска, ознакомление их с современными методами позна-

ния. На основе содержания принципа научности может быть выделен ряд требований к КОП. Прежде всего, *обучающие программы целесообразно наполнять таким содержанием, которое наиболее эффективно может быть усвоено только с помощью компьютера*. В первую очередь это касается демонстрации процессов, сущность которых может быть выявлена только на микроуровне организации вещества; процессов и явлений, реализация которых в условиях школьного эксперимента невозможна по ряду причин: трудности его параметров, опасности проведения опытов в лаборатории, длительности или, напротив, мгновенности его протекания и т. д., т. е. там, где надо заменить реальный эксперимент.

Несмотря на то что следующее требование формулируется весьма тривиально — *содержание КОП должно соответствовать современному состоянию научного знания*, роль его значительна. Это подтверждается анализом ряда программ, составители которых довольно смело оперируют неверными, неточными, а порою и вымышленными условиями и параметрами моделируемых процессов.

Наконец, способы усвоения учебного материала, предусмотренные программой, должны быть адекватны современному научным методам познания. В качестве последних могут быть названы метод моделирования, в том числе математического, метод системного анализа, способствующие более глубокому познанию объектов, представляющих собой сложно организованные системы.

Традиционная трактовка принципа *наглядности*, как известно, сводится к тому, чтобы создать у учащихся чувственное представление об изучаемом объекте. Достижения психологической и педагогической науки позволили в последние годы уточнить и дополнить его. Содержание этого принципа уточняется введением дополнительного требования. Согласно этому требованию, выполнение деятельности учащимися должно протекать в перцептивном плане. Подобное понимание принципа наглядности требует точного указания тех действий, которые необходимо выполнить с предметами, чтобы, с одной стороны, выявить содержание будущего понятия, а с другой — представить это содержание в виде знаковых моделей. На основании данного принципа можно сформулировать следующие требования к обучающим программам: *в КОП при отражении чувственного объекта не следует увлекаться натурализмом; в программе должна быть представлена не любая модель, а только та, которая способствует реализации дидактических целей данной обучающей про-*

граммы; модель, содержащуюся в программе, следует представлять в форме, позволяющей наиболее четко раскрыть существенные связи и отношения объекта; существенные признаки, связи и отношения модели должны быть в программе адекватно зафиксированы цветом, миганием, звуком и т. д.

Наиболее важное требование, основывающееся на современном понимании принципа наглядности, состоит в том, что с помощью КОП необходимо не только представлять объект изучения, но и организовывать деятельность учащихся по его преобразованию. В современных исследованиях [5] показано, что эффективность обучения повышается, когда учащиеся сами строят модели, а не получают их в готовом виде. Действительно, строя модели, школьники воспроизводят признаки объекта, его структуру. В процессе построения дальнейшей работы с моделью усваиваются теоретические знания, создается возможность для самостоятельного выведения знаний при переходе на аналогичный материал. Работа с моделью может вестись в двух направлениях: достраивания модели или ее видоизменения и переконструирования. Деятельность учащихся при работе с моделью должна быть заранее сформирована, только в этом случае работа с компьютерной программой моделирующего типа может быть эффективна.

На основе деятельностного подхода пересмотрено также содержание принципа систематичности и последовательности. Если ранее он предполагал только систематизацию знаний, в их отрыве от деятельности по усвоению этих знаний, то в настоящее время в содержание этого принципа вводится второй компонент — представление адекватных действий, в которых эти знания усваиваются, что является важным дополнением к его традиционному содержанию. Чтобы у учащихся с самого начала сложилась система представлений о планируемой деятельности, необходимо на начальном этапе обучения дать общее предписание (ориентировочную основу действия) для последующей работы. Такая ориентировка позволяет ликвидировать разрыв между систематическим представлением изучаемого материала и умением применять полученные знания при решении практических задач.

Существует точка зрения, что принцип систематичности в его традиционной трактовке должен быть заменен на принцип системности. Содержание последнего трактуется как требование формирования у учащихся знаний, адекватных структуре научной теории. В настоящее время показано, что требование системности не отменяет и не компенсирует дидактического принципа систематичности. Последний должен включать в себя в дидактически преломленном виде идеи, связанные с общенаучным методом системности.

Так, требование дидактического принципа систематичности следовать в обучении логике науки обогащается, согласно требованию системности, следующим положением: содержание учебного предмета должно отражать не просто логику науки, а логику, адекватную ее современному состоянию, — логику системного раскрытия объектов и явлений изучаемой действительности.

Данная трактовка принципа систематичности и последовательности позволяет наметить ряд требо-

ваний к обучающим программам, в том числе и компьютерным. Наряду с предметными знаниями в содержание программ должны войти и специальные методологические знания, отражающие структуру соответствующей науки.

В состав методологических знаний целесообразно включать системные методы познания. Эти методы могут быть наиболее оптимально реализованы с помощью КОП, отсюда: в объектах или явлениях, представляемых с помощью компьютерных программ, должны быть выделены основные структурные элементы и существенные связи между ними, позволяющие представлять этот объект (явление) в виде целостного образования. Наряду с этим, алгоритм, в соответствии с которым строится деятельность обучаемого по усвоению материала, должен отражать логику его системного анализа.

По мнению Н. Ф. Талызиной, принцип активности должен «...не просто призывать учителя к организации деятельности учащихся, но и указывать критерии выбора наиболее рациональных ее видов» [6, с. 40]. Таким критерием, прежде всего, является адекватность содержания действий учащихся усваиваемым знаниям и нормативным действиям, причем активность выступает теперь как требование не только воспроизведения учащимися предметных действий, но и собственно учебных действий, в ходе которых происходит и усвоение предметных умений.

Отсюда следует, что содержание деятельности, организуемой с помощью КОП, должно соответствовать усваиваемым знаниям. Так, если программа нацелена на формирование у учащихся типовых умений, то организацию деятельности следует вести по готовому алгоритму, заданному составителем программы. Напротив, если предполагается формирование умения решать эвристические задачи, то необходимо предоставить обучаемым возможность самостоятельно построить алгоритм действий. Компьютер в подобной ситуации выступает лишь в роли «контролера», «эксперта».

Поскольку активность обусловлена сознательностью, при разработке КОП необходимо ориентироваться на следующие требования: целесообразно в структуру обучающей программы вводить ориентировочный компонент, который должен включать два вида знаний — знание о деятельности, реализуемой с помощью КОП (цель деятельности, ее предмет, средства и основные этапы осуществления), и предметные знания, необходимые для успешной работы с программой (формулы, правила, справочно-информационные данные и т. п.).

Традиционная трактовка принципа индивидуального подхода в обучении была ориентирована лишь на отдельные характеристики учащегося и на соответствие учебного материала этим характеристикам. В современном понимании принцип индивидуализации обучения базируется на идеях целостного, личностного подхода к учащемуся, как к субъекту деятельности [8]. Его содержание предстает как система индивидуализированных способов и приемов взаимосвязанных действий преподавателя и учащихся, которая органично присуща всем этапам процесса обучения и направлена на всесторонний учет индивидуальных особенностей учащихся. В таком понимании данный

принцип включает в себя требования учета и последующего формирования мотивационного, эмоционально-волевого и интеллектуального компонентов личности учащегося.

Совокупность требований к обучающим программам, реализующим принцип индивидуализации обучения, может быть представлена следующим образом: *учет при отборе и построении учебного материала, при выборе методики его усвоения, исходного уровня предметных и учебных знаний и умений учащегося, а также уровня развития мотивационной сферы, индивидуально-личностных, психофизиологических особенностей каждого учащегося.* Программы компьютерного типа, в отличие от традиционных, позволяют уже сегодня в значительной степени реализовать указанные требования.

На основе деятельностного подхода рассмотрено и содержание принципа *доступности*. Доступность понимается как возможность достижения цели обучения. Условием ее реализации является наличие к началу обучения всех его внутренних и внешних условий. Полная и обобщенная система этих условий предполагает наличие:

субъекта учения, т. е. учащегося, обладающего необходимыми предметными и учебными знаниями и умениями, системой положительных учебных мотивов, психофизическими предпосылками для успешного осуществления учения и находящегося в состоянии актуальной готовности к выполнению этой деятельности;

учебного материала, соответствующего указанным выше характеристикам учащегося;

адекватной методики усвоения, т. е. системы методов, приемов, средств обучения, соответствующей специфике данного учебного материала, а также отмеченным характеристикам учащегося;

благоприятных внешних условий осуществления деятельности учения.

Принцип доступности при таком содержании приобретает обобщающий характер по отношению к другим дидактическим принципам, отражающим его процессуальную сторону (принципы наглядности, систематичности и последовательности, индивидуализации обучения и пр.). Если последние направлены на создание *отдельных* условий, то принцип доступности призван констатировать наличие *системы* этих условий. При одновременном наличии названных выше условий обучения можно говорить о реализованной доступности, т. е. о возможности достижения цели.

В таком понимании принцип доступности распространяется на все структурные элементы деятельности учения в их системе. Обучающая компьютерная программа выступает лишь в качестве одного из структурных элементов этой системы,

поэтому доступность цели (усвоение этой программы) может быть обеспечена лишь в том случае, если к началу осуществления очередного акта обучения (фрагмента совместной деятельности преподавателя и учащихся), в который включена обучающая программа, будет представлена вся совокупность внутренних и внешних условий обучения.

Таким образом, для достижения цели обучения (усвоения обучающей программы) к началу осуществления фрагмента обучения, в состав которого включена обучающая программа, необходимо иметь в наличии указанные выше внутренние и внешние условия.

Мы рассмотрели лишь некоторые требования к КОП, не претендуя на их полное описание, так как решение этой задачи связано с дальнейшим исследованием проблемы дидактических принципов — оптимизацией содержания каждого из них и обоснованием их системы.

Литература

1. *Габай Т. В., Чернявская А. Г.* Педагогический принцип доступности в свете деятельности теории // Вестник Московского университета. 1987.
2. *Кирсанов А. А.* Индивидуализация учебной деятельности как педагогическая проблема. Казань, 1982.
3. *Милорадова Н. Г.* Содержание и функции наглядности при разных типах учения: Автореф... дис. канд. психол. наук. М., 1980.
4. *Резницкая Г. И.* Зависимость реализации принципа сознательности от характера предмета усвоения: Автореф... дис. канд. пед. наук. М., 1984.
5. *Салмина Н. Г.* Виды и функции материализации в обучении. М., 1981.
6. *Тальзина Н. Ф.* Педагогика и психология: Круглый стол // Вопросы психологии. 1981. С. 39—40.
7. *Терегулов Ф. Ш.* Принцип наглядности и материализации деятельности в процессе обучения: Автореф... дис. канд. пед. наук, М., 1979.
8. *Философско-психологические проблемы развития образования / Под ред. В. В. Давыдова. М., 1981.*
9. *Чернышева В. К.* Принцип систематичности и последовательности при построении содержания обучения: Автореф... дис. канд. пед. наук. М., 1980.

Н. ЮНЕРМАН
г. Новосибирск

Использование графической системы ШПАГА при введении основных понятий программирования

52 В ВЦ СО АН СССР разработан пакет графических процедур ШПАГА, предназначенный для использования при обучении младших школьников. Каждая процедура имеет два названия — полное и сокращенное. Рисующим инструмен-

том является луч на экране монитора.

В версии графической системы (описания различных версий системы см. в работах 1—4), реализованной на ПЭВМ «Агат», имеются следующие процедуры.

Процедура

ПОДВОД (X, Y)

П (X, Y)

ЛУЧ (X, Y)

Л (X, Y)

ТОЧКА (X, Y)

Т (X, Y)

ОКРУЖНОСТЬ (X, Y, R)

ОКР (X, Y, R)

ДУГА (X1, Y1, X2, Y2, X3, Y3)

Д (X1, Y1, X2, Y2, X3, Y3)

НАДПИСЬ (ТЕКСТ, X, Y)

НП (ТЕКСТ X, Y)

Ц (Т)

ПРЯМОУГОЛЬНИК (X1, Y1, X2, Y2)

ПР (X1, Y1, X2, Y2)

ТРЕУГОЛЬНИК (X1, Y1, X2, Y2, X3, Y3)

Выполняемое действие

Подводит луч в точку с координатами (X, Y).

Проводит отрезок прямой из текущей точки в точку с координатами (X, Y).

Изображает точку с координатами (X, Y).

Изображает окружность радиуса R с центром в точке (X, Y). После завершения рисования точка (X, Y). Изображает дугу окружности, соединяющую точку (X1, Y1) с точкой (X3, Y3) и проходящую через точку (X2, Y2). Точка (X3, Y3) становится текущей.

Вычеркивает надпись, заданную значением имени ТЕКСТ, начиная с точки (X, Y).

Устанавливает цвет рисунка. Значения для имени Т: «БЕЛЫЙ» или «Б» «КРАСНЫЙ» или «К» «ЖЕЛТЫЙ» или «Ж» «ЗЕЛЕНый» или «З» «СИНИЙ» или «С» «ФИОЛЕТОВЫЙ» или «Ф» «ЧЕРНЫЙ» или «Ч». Рисует закрашенный прямоугольник, противоположные вершины которого находятся в точках (X1, Y1) и (X2, Y2).

Рисует закрашенный треугольник с

ТР (X1, Y1, X2, Y2, X3, Y3)

КРУГ (X, Y, R)

К (X, Y, R)

СЕКТОР (X1, Y1, X2, Y2, X3, Y3)

СК (X1, Y1, X2, Y2, X3, Y3)

СЕГМЕНТ (X1, Y1, X2, Y2, X3, Y3)

СГ (X1, Y1, X2, Y2, X3, Y3)

МНОГОУГОЛЬНИК (X, Y, R, N)

МНГ (X, Y, R, N)

ОБЛАСТЬ (X, Y)

ОБЛ (X, Y)

вершинами в точках (X1, Y1), (X2, Y2) и (X3, Y3).

Рисует закрашенный круг радиуса с центром в точке (X, Y)

Изображает закрашенный сектор, вырезанный из круга с центром в точке (X1, Y1); (X2, Y2) и (X3, Y3) — координаты концов дуги, стягивающей сектор.

Изображает окрашенный круговой сегмент, границами которого служат дуга, проведенная из точки (X1, Y1) в точку (X3, Y3) через точку (X2, Y2) и отрезок прямой, соединяющий концы этой дуги (хорда). Изображает правильный N-угольник с горизонтальным основанием; X, Y, R — параметры описанной окружности.

Закрашивает текущим цветом все точки, расположенные в окрестности точки (X, Y) и внутри ближайшего замкнутого контура.

53

Рассмотрим последовательность введения основных понятий программирования на примере решения задач, связанных с построением изображений. Изучению этой темы предшествовало знакомство учащихся с понятием имени в языке программирования, с понятиями «предписание присваивания», «система координат».

Приведенные ниже методические рекомендации предназначены для работы с учащимися IV—V классов.

Знакомство с графической системой, на наш взгляд, следует начинать с изучения процедур ПОДВОД, ЛУЧ, Ц и ТОЧКА. Предоставьте каждому ученику возможность нарисовать картинку в соответствии с его желанием. Затем предложите задания по составлению рисунков, удовлетворяющих определенным условиям. Например:

составить рисунок с повторяющимися элементами (нотный стан, лестница, елочка, многоэтажный дом, различные орнаменты);

выполнить один и тот же рисунок разными цветами;

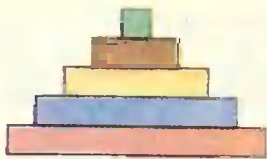
один и тот же рисунок расположить в разных частях экрана.

Выполнение этих заданий (их целе-

сообразно выполнять и при последующем изучении других процедур системы) подведет учащихся к усвоению таких понятий программирования, как «цикл» и «процедура». После того, как учащиеся освоили работу с линейными процедурами, перейдем к изучению простейших растровых процедур, например, процедуры ПРЯМОУГОЛЬНИК. Выполнив тренировочные упражнения — рисование разноцветных прямоугольников любых размеров, — учащиеся составляют рисунки, содержащие прямоугольники, расположенные независимо друг от друга. Затем, постепенно усложняя задание, предложим ребятам составить рисунок, содержащий прямоугольники, расположение которых взаимосвязано. При этом важно обсудить с учащимися последовательность изображения геометрической фигуры. Можно предложить детям составить картинку только из прямоугольников (рис. 1, 2).

Последовательность упражнений при изучении процедуры ТРЕУГОЛЬНИК — такая же, как и при изучении процедуры ПРЯМОУГОЛЬНИК. Завершим серию упражнений заданием на составление рисунков, содержащих только треугольники и прямоугольники (рис. 3). После изучения обеих процедур, целесообразно

1



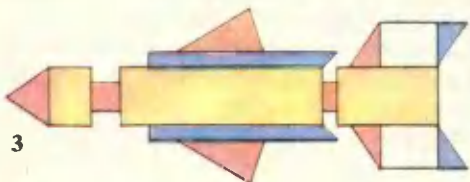
2

предложить ребятам вопрос: представьте себе, что вам разрешено использовать только одну из процедур — либо ПРЯМОУГОЛЬНИК, либо ТРЕУГОЛЬНИК. Какую из них вы бы предпочли? Почему?

При изучении процедур ОКРУЖНОСТЬ и КРУГ в первую очередь обратите внимание ребят на различие этих геометрических фигур. Затем учащиеся могут выполнить такие задания: составить рисунки только из кругов (снежная баба, воздушные шары, пуговички), кругов и прямоугольников, кругов и треугольников, всех рассмотренных фигур.

Размещая один и тот же рисунок в разных частях экрана, важно обсудить с ребятами, чем отличаются программы для рисования, что общего между ними. При этом в каждом рисунке выбрать одну так называемую узловую, точку и координаты всех остальных определять через ее координаты (через прира-

3



щения). Если при этом программу оформить в виде процедуры, в начале которой выполнить присваивания значений координатам узловой точки, то, изменяя эти значения, получим изображения одного и того же рисунка в различных частях экрана. Теперь можно ввести понятие процедуры с параметрами, выбрав в качестве параметров координаты узловой точки. Введя переменную для обозначения какого-нибудь размера и придавая ей различные значения, получим подобные изображения (например, парашютный десант, улицу из одинаковых домов, аллею деревьев и т. д.). При выполнении всех рисунков предложите учащимся варьировать цветом, подбирая наиболее удачные сочетания.

Для иллюстрации понятия «процедура» рассмотрите задачу рисования дорожных знаков, разбив их на группы и выделив в них одинаковые элементы.

С помощью процедуры ОБЛАСТЬ можно получать закрашенные изображения различных объектов. Для этого необходимо сначала нарисовать контур нужного объекта и воспользоваться процедурой ОБЛАСТЬ, выбрав в качестве ее параметров координаты любой точки, расположенной внутри нарисованного контура.

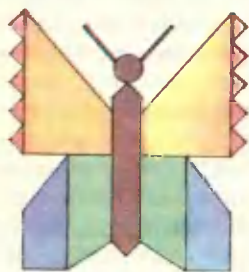
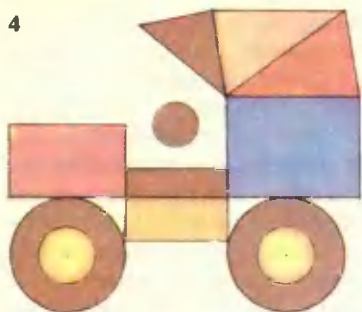
Изучение процедур СЕГМЕНТ и СЕКТОР поможет в усвоении соответствующих понятий в курсе математики.

Можно провести конкурсы по составлению рисунков с заданным условием или на определенную тему, например: составить узор для вышивки, паркета, аппликации и др.

Завершая изучение этой темы, предложите ребятам написать программы выполнения рисунков, для изображения элементов которых нет специальных процедур. Такие задания требуют творческого подхода: нужно определить, в какой последовательности и какие процедуры должны быть использованы, чтобы получить заданный рисунок. Примерами таких заданий служат рисунки 4, 5.

Еще один вид заданий — составление программ для рисования симметричных картинок, например, снежок. В этом случае можно написать процедуру с параметрами для рисования части картинка,

4



5

а затем записать последовательность вызовов этой процедуры для рисования всей картинке. Процедура ПОВОРОТ, которая поворачивает систему координат на заданный угол, упростит решение таких задач.

При выполнении заданий необходимо обращать внимание учащихся на составление плана (алгоритма) рисования и выбор оптимальных решений (например, сначала нарисовать все одним цветом, потом другим и т. д.). В тех случаях, когда ребята затрудняются составить рисунок с определенными свойствами,

им нужно предложить написать программу к уже имеющемуся рисунку или «разукрасить» уже имеющийся рисунок.

Если в вашем распоряжении есть другая графическая система, то воспользовавшись данной методикой введения основных понятий программирования, выберите соответствующее подмножество процедур и порядок их изучения.

При отсутствии какой-либо графической системы можно выполнять все рассмотренные упражнения «вручную», на бумаге. Как показывает опыт, и в этом случае ребята проявляют большой интерес к изучаемому материалу. Для наглядности приготовьте небольшой конструкторский набор, состоящий из треугольников, прямоугольников и кругов различных цветов и размеров.

55

Литература

1. Звенигородский Г. А. Первые уроки программирования / Под ред. А. П. Ершова. М., 1985. 208 с.

2. Звенигородский Г. А., Салихова А. К., Цикола В. А. Машинная графика в математическом обеспечении учебного процесса // Прикладные методы информатики.— Новосибирск, 1980. С. 121—133.

3. Салихова А. К., Соколова Н. А. Графическая система ШПАГА // Квант. 1980. № 1. С. 56—59.

4. Салихова А. К., Соколова Н. А. Комплекс процедур машинной графики ШПАГА. Новосибирск. 1979. 12 с.— (Препринт / АН СССР. Сиб. отд.-ние ВЦ; 169).

С. КОЗЛОВСКИЙ

Узловые проблемы оценки качества изображения на экране дисплея

При работе с персональными ЭВМ от операторов и учащихся, длительное время пользующихся дисплеями, поступают жалобы на боль в глазах, потерю остроты зрения, головные боли, утомляемость, головокружение, сердечно-сосудистые заболевания. Есть предположения также, что отрицательное влияние работы за дисплеем на здоровье тем

сильнее, чем моложе оператор. Ряд авторов отметили, что условия, вынуждающие операторов смотреть с напряжением, могут вести к указанным болезненным ощущениям. Особо подчеркивалось то влияние, которое оказывает на утомление уровень (состояние) качества изображения. Это положение было подтверждено экспериментально в совмест-

ном комплексном исследовании, выполненном в психологических лабораториях университетов Оксфорда, Майями, Питсбурга, Огайо и др.

В этом исследовании была поставлена задача сравнить влияние типа дисплея на утомление оператора. Объектом сравнения служили дисплеи с электронно-лучевой трубкой (ЭЛТ) и его макет-копия. Испытуемые должны были найти определенную букву в одной из четырех колонок, изображенных на дисплее. Каждая колонка состояла из 20 слов. Данные об ЭЛТ-раздражителях были заложены в компьютер, а данные о раздражителях копии напечатаны на стандартных листах бумаги и помещались на вертикальном пюпитре.

56

ЭЛТ-символы (по сравнению с копией) были расплывчаты и имели мерцания малых площадей. Необходимость правильной фокусировки на относительно расплывчатых символах приводила к непрерывному действию хрусталика глаза. Мерцание малых площадей также требовало постоянной регулировки нервных рецепторов глаз, приспособляющихся к меняющимся условиям освещения. Это усиливало нагрузку зрительного механизма, что способствовало дискомфорту и утомлению.

Одним из показателей утомления авторы избрали среднюю частоту морганий в минуту. Установлено, что при работе с ЭЛТ-дисплеем частота морганий в течение двух часов увеличивалась почти в полтора раза, а при работе с копией частота морганий оставалась стабильной.

Был проведен сравнительный анализ влияния двух моделей дисплеев на зрительное напряжение (критерий влияния — рост числа жалоб операторов). Более новая модель обладала лучшей контрастностью, большим размером знаков, а также особым экраном, который устранял блики. Установлено, что число жалоб на зрительное напряжение у лиц, работающих на новых моделях дисплеев по сравнению со старыми, уменьшилось вдвое.

Напомним, что в работах советских психологов также установлено влияние уровня освещенности и отклонения от оптимального контраста на общую работоспособность оператора (5).

Из сказанного выше следует, что повышение качества изображения на экране дисплея — важная задача сохранения работоспособности школьников, обучающихся за экраном дисплея.

По мнению зарубежных специалистов, еще отсутствуют экспериментальные данные, которые могли бы связать специфические характеристики видеотерминала с симптомами зрительной усталости. Это означает, что на возникновение таких симптомов заболевания особое влияние оказывает интегральное качество изображения. Некоторые исследователи отмечают также влияние единичных параметров качества изображения (отклонение от оптимального контраста, резкости и других параметров на состояние здоровья и работоспособность пользующихся дисплеем.

Отсюда возникает необходимость измерения единичных и интегральных показателей качества изображения на экранах дисплея. Это дает возможность установить допустимые пределы отклонений от оптимального качества изображений и не внедрять в школе дисплеи, которые не соответствуют хорошему качеству изображения.

Величины некоторых единичных показателей качества измеряются инструментальными (физическими) методами. К ним относятся яркость, число градаций яркости, контраст, число мельканий на ширину экрана, расплывчатость (резкость) изображений, число цветов. Некоторые показатели не могут быть измерены инструментальными методами. К ним относится интегральное качество изображения. Для оценки этого параметра необходима разработка иных методов оценки, более точно характеризующих особенности человеческого восприятия.

Вместе с тем некоторые эргономические вопросы могут быть решены только на основе сопоставления результатов измерений, полученных инструментальными и психометрическими методами. Неинструментальные методы измерений могут быть объективными и субъективными. В ряде отраслей промышленности (полиграфия, легкая и пищевая промышленность, ювелирное дело) в качестве объективной меры ис-

пользуются каталоги качества продукции. Они представляют собой ряды предметных эталонов состояний качества, построенных по определенным признакам (показателям).

Особенно важно создать каталоги для оценки интегрального качества изображений, а также такого сложного субъективного показателя, как ясность, сочность изображений. Уровень (состояние) качества последнего, его субъективная оценка определяется несколькими единичными физическими показателями: резкость (размытость), контрастность — отношением сигнал — помеха, яркость. И если физические значения этих показателей могут быть измерены инструментально, то оценка их совокупного влияния на ясность, сочность изображений требует иных методов измерений, например каталогов качества и психометрических измерений.

Обратим внимание на то, что понятие «интергальное качество изображения» шире понятия «ясность, сочность, изображения», поскольку интегральное качество включает в себя и геометрические формы, отклонения от них (искривление прямых линий, превращение круга в эллипс и т. п.). Элементом психометрических измерений является субъективная оценка, которая выражается в баллах или рангах. Устанавливается число таких ступеней оценки. Оно колеблется от 2 до 100 баллов или рангов. Устанавливается также порядок и частота предъявления уровня качества изображения (4).

Субъективные методы оценки качества изображений разрабатываются на основе психометрических методов измерений, наиболее адекватно отражающих эту оценку. При использовании указанных методов эксперт сам оценивает качество изображения. Субъективная оценка — элемент, входящий в некоторую систему психометрических измерений. Она включает в себя помимо того еще несколько составляющих: тип шкалы, число рангов или баллов, определение уровня качества изображения; заметность — незаметность искажений; отлично — неудовлетворительно, лучше — хуже, подробное описание ка-

чества, методы математической обработки (4).

Необходимо отметить, что психометрические методы являются ведущими при оценке качества изображения (2). Ведущими в том смысле, что инструментальные измерения могут применяться только в случае, если их результаты коррелируют (коэффициент корреляции близок к единице) с результатами психометрических измерений. Например, размер светящейся точки на экране или величина яркости перепада на границе знака, измеряемые инструментально, с одной стороны, и субъективная оценка резкости (размытости) изображения — с другой.

Необходимо отметить, что величина коэффициента корреляции между результатами этих двух видов измерений резкости изображения (так же как и других параметров качества) не всегда приближается к единице. Это связано с тем, что на субъективную оценку резкости (размытости) изображения оказывают влияние такие параметры, как контрастность, яркость, отношение «сигнал — шум». Это влияние учитывается при измерении качества изображения психометрическими методами. В то же время инструментальными методами физическая резкость измеряется без учета этих составляющих. Этим объясняется отсутствие корреляционной связи между результатами измерений, проведенных инструментальными и психометрическими методами, одного и того же параметра (показателя) качества.

Сопоставление результатов психометрических и инструментальных измерений применяется также для решения других задач. Так устанавливаются оптимальные значения показателей качества (единичных и целостного). Оптимальным будет состояние, получившее в процессе психометрических измерений высший балл.

Другая существенная характеристика показателя качества, которая устанавливается по результатам психометрических измерений, — допустимая величина отклонений от его оптимального значения. По данным отечественных и зарубежных исследований (4), указанная величина равна 3,75 балла.

Состояния (уровни) качества, которые оцениваются баллами 5 (оптимальное значение) и 3,75 (допустимое значение), измеренные также инструментально, позволяют установить физические величины, соответствующие этим баллам. Вместе с тем для определения оптимального качества изображений и допустимых значений отклонений от оптимального качества на экране школьного дисплея необходимо использовать не только психометрические, но и также инженерно-психологические (число ошибок, например) и санитарно-гигиенические методы исследований.

При исследовании качества изображения возникают и такие проблемы.

58

Если два или несколько изображений имеют разные физические величины яркости (B_1, B_2, \dots, B_n), контрастности (K_1, K_2, \dots, K_n), резкости ($R_1, R_2, R_3, \dots, R_n$) и все эти величины находятся в пределах допустимых значений, то сравнительную оценку интегрального качества этих изображений (Q_1, Q_2, \dots, Q_n) можно установить, используя два метода.

Первый — установление целостной оценки. Это делается на основе психометрических измерений. Субъективная оценка относится здесь к целостному, интегральному качеству изображения. Второй метод — измерение интегрального качества изображения на экране дисплея на основе комплексной оценки качества. Комплексная оценка строится с применением определенного алгоритма, измерения единичных показателей качества с учетом их весомостей, т. е. вкладов каждого показателя в комплексную оценку.

Аналитически комплексная оценка может быть выражена функциональной зависимостью:

$$Q_{\text{комп}} = f(k_1 X_1, k_2 X_2, \dots, k_n X_n),$$

где X — показатели качества изображений, измеряемых инструментально, k — коэффициенты весомости, которые могут носить и характер функции,

т. е. изменяться с изменением величины единичного показателя. k устанавливается психометрическими методами.

Целостная оценка качества изображения формулируется субъектом (экспертом) на основе опыта и системы эталонов (образцов), отражающих состояния (уровни) качества. Ее значение велико для потребителя, в нашем случае для ученика, обучающегося на школьном дисплее.

Комплексная оценка имеет существенное значение для разработчиков дисплеев, поскольку базируется на основе инструментально измеряемых показателей качества, значениями которых разработчик может управлять, учитывая их оптимальные характеристики.

Узловые оценки качества изображений, формируемых на экране дисплея, являются:

совершенствование инструментальных и психометрических методов измерений; установление связей между результатами физических и психометрических измерений;

установление оптимального значения и допустимых пределов отклонений от оптимального качества;

создание каталогов для более тонкой оценки интегрального качества изображения.

Литература

1. Козловский С. М. Методы зонирования кинозалов // Техника кино и телевидения. 1973. № 3. С. 9—14.
2. Субъективная оценка качества кино- и телевизионных изображений. Ее связь с объективными измерениями / Под ред. С. М. Козловского. М., 1970.
3. Козловский С. М. Психофизические проблемы оценки качества промышленных изделий // Вопросы психологии. 1977, № 6. С. 60—69.
4. Козловский С. М. Современные психометрические методы оценки качества кино- и телевизионных изображений. М., 1978.
5. Литвак И. И., Ломов Б. Ф., Соловейчик И. Е. Основы построения аппаратуры отображения в автоматизированных системах. М. 1975.

Структура и функции обучающей системы

Разработка учебных программных средств, реализующих различные педагогические функции, — одна из важнейших задач компьютеризации обучения. В ближайшем будущем учителю и учащимся в течение одного урока придется работать с набором учебных программ. Эти программы должны быть объединены в систему, определяемую целями урока.

Представляемая в данной статье система «Ускорение» * является для учителя средством подготовки и проведения урока с использованием компьютера; для учащихся — инструментом познавательной деятельности (как носитель набора различных по своим функциям обучающих программ); для программиста — средством разработки обучающих программ.

В систему входят четыре компонента: библиотека обучающих программ, реализующих различные дидактические функции; редактор текстов, при помощи которого преподаватель составляет или корректирует сценарий урока; монитор, обеспечивающий выбор и реализацию сценария урока. Инструментарий программиста, состоящий из графического редактора и базы знаний (по химии, VII класс).

Ценность системы как средства обучения определяется набором программ (главным образом программы моделирования и программы-тренажеры).

Программы моделирования имитируют химические и химико-технологические процессы, дают возможность учащимся активно «вмешиваться» в изучаемый процесс, «управлять» им, видеть результаты своей деятельности на экране дисплея.

В качестве примера назовем программному моделирующего типа «Образование воды из газообразных водорода и кисло-

рода» по теме «Первоначальные химические понятия». Этот тип программ целесообразно использовать в тех случаях, когда необходимо организовать усвоение таких процессов и явлений, демонстрация которых в условиях реального эксперимента в школе невозможна.

Программы-тренажеры нацелены на отработку у учащихся типовых умений. В таких программах фиксируется четко сформулированная цель работы, представляется задание, показывается образец деятельности. В программах предусмотрены контроль операций и в случае необходимости коррекция действий обучаемого. Комментарии и реплики носят диалоговый характер. Программы позволяют работать в индивидуальном режиме. Учащийся может пользоваться одной и той же программой (в рамках допустимого учебного времени) до тех пор, пока не удостоверится в своем успехе.

Примером данного типа программ является программа «Составление химических уравнений». К этому типу относятся также программы, ориентированные на отработку навыков решения расчетных задач. Решение задач с помощью обучающих программ имеет ряд достоинств, таких, как коррекция действий обучаемого, перевод учащегося (в случае необходимости) с одного уровня сложности на другой.

Рассмотрим подробнее деятельность учителя при подготовке урока с помощью системы «Ускорение». При помощи стандартного редактора учитель создает сценарий урока, используя программы из библиотеки. Сценарии уроков хранятся в файле BOOK. Инструкция с правилами составления сценария не подлежит корректировке и носит информационный характер. Первое оглавление содержит названия тем курса. Второе состоит из названий уроков по каждой теме. Оба оглавления можно дополнять и корректировать. Далее по выбранной теме урока разрабатывается сценарий, который со-

* Система разработана в двух лабораториях: информатики и вычислительной техники НИИ СиМО АПН СССР и представления знаний ИПС АН СССР.



Структурная схема системы «Ускорение»

стоит из поясняющих текстов и названий обучающих программ. Например, в поясняющих текстах, составляемых учителем, можно указать цель урока, методы его проведения, перечень умений, которые должны быть сформированы в ходе работы с программами, и т. д. Названия обучающих программ берутся из оглавления библиотеки. Библиотека может пополняться.

Для подключения программ в сценарии необходимо набрать с первой позиции строку вида @ <имя программы>.

Выделим совокупность педагогических функций, которые могут быть реализованы в сценариях уроков, составленных с помощью системы «Ускорение», и определим их место в учебном процессе.

Первая группа функций связана с организацией усвоения материала. В сценарии можно организовать отработку (закрепление) знаний и умений обучающихся.

Вторая группа функций — контролирующая. Система позволяет осуществлять диагностику исходного уровня знаний и умений учащихся, их психофизиологических особенностей; про-

водить текущий и тематический контроль с выдачей итоговой информации.

Третья группа функций связана с демонстрацией различных объектов и явлений изучаемой действительности как в статическом (текст, графика, диаграмма и т. д.), так и в динамическом виде.

Система может выступать эффективным инструментом в деятельности учащихся как на уроке, так и во внеурочное время.

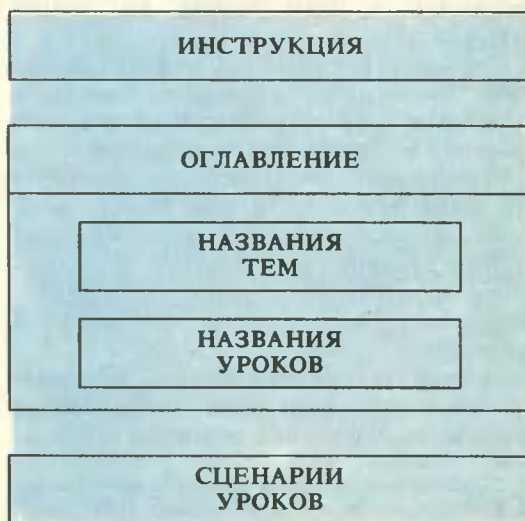
Работа учащегося с системой заключается в следующем: ученику выдается оглавление тем уроков, далее с помощью курсора выбирается нужная тема. Затем учащемуся выдается оглавление уроков по данной теме, в котором он выбирает нужный урок. Аналогично выбирается и сценарий урока. Дальнейшие действия системы обусловлены этим сценарием.

Эффективность использования системы определяется реализацией указанных педагогических функций.

В большинстве программ предусмотрена обработка ответов учеников и обеспечена комфортность общения.

Для повышения производительности труда программиста при создании обучающих программ система снабжена специальным инструментарием.

Инструментальные средства программиста состоят из графического редактора, кадротеки и базы знаний (см. схему).



Структура файла BOOK

Графический редактор VGRAF разработан в ВЦ АН СССР и предназначен для оформления информации на экране дисплея. С его помощью легко разрабатывать кадры для обучающих программ, составленных на любом языке. Среднее время восстановления экрана — одна секунда. С помощью редактора создается кадровая обучающих программ. VGRAF наиболее целесообразно использовать при разработке программ моделирующего типа.

База знаний по химии реализована на языке micro-PROLOG. Здесь представлены следующие понятия: химические элементы, простые вещества, кислотные остатки, оксиды, гидрооксиды, кислоты, соли, а также реакции: замещения, обмена, нейтрализации, соедине-

ния, разложения, окислительно-восстановительные реакции; можно получить ответ на любой запрос, сформулированный в терминах приведенных понятий. В базе знаний содержатся также описания ошибок, которые могут быть в запросах. Разработан интерфейс базы знаний с языком Паскаль. База знаний позволяет существенно облегчить процесс написания программ типа тренажер.

Таким образом, мы описали обучающую систему, которая выступает в трех основных аспектах: как средство обучающей деятельности педагога, как средство познавательной деятельности учащегося и как средство написания обучающих программ для программиста.

И. БОБКО, А. ПЛОТНИКОВ, Г. САПРЫКИНА
НИИИиВТ АПН СССР

КУВТ-86 в подготовке работников народного образования

С 13.07.87 по 18.07.87 в Новосибирске на базе НИИИиВТ был проведен семинар с участием представителей республиканских институтов усовершенствования учителей по методике использования 36-часового спецкурса по основам информатики и вычислительной техники для руководящих и педагогических работников народного образования на базе комплекса КУВТ-86.

Курс ориентирован на лиц, впервые приступивших к изучению информатики и освоению вычислительной техники, с тем чтобы они в короткий срок получили навыки практической работы на ПЭВМ «Электроника БК-0010». Тематический план и программа курса опубликованы в журнале «Информатика и образование», № 2 за 1986 г. План включает семь тем, машинная поддержка которых обеспечивается набором программ (см. табл.).

Программно-методический комплекс состоит из трех компонентов:

пакета прикладных программ, включа-

ющего 34 игровые, демонстрационные и обучающие программы;

руководства учителя, содержащего тематический план, описание назначения используемых программ, лекционный материал;

руководства слушателя, содержащего основные понятия тематического плана, подробные инструкции для работы с каждой программой.

Для рассылки программ на рабочее место слушателя с рабочего места преподавателя используется система «Рига», имеющая четыре сервисные функции. После ее загрузки в память ДВК-2М на экране РМП появляется меню, содержащее 4 варианта:

- 1) передача по сети;
- 2) прием с БК-0010;
- 3) печать;
- 4) каталог.

Работу следует начинать с пункта 4 (каталог). При вводе числа 4 система спрашивает, каталог какой дискеты (с какой щели дисковода) нужно распе-

Содержание темы	Количество часов			Имена используемых программ
	Всего	Лекции	Практические занятия	
Знакомство с ЭВМ	6	—	6	KOSAR.FOC AUTO.RIG SARAN.RIG LOGIC.FOC CHESS.RIG OKEAN.RIG LUNA.FOC KLAW.RIG ROBOT.RIG REDAKT.RIG
ЭВМ в современном обществе. Обработка текстов	5	1	4	ARHIV.RIG ELTA64.FOC ELTA32.FOC BD64.FOC BD32.FOC
ЭВМ в современном обществе. Информационные системы	3	1	2	LEGKIE.FOC SERDS.FOC TERPEN.FOC
ЭВМ в современном обществе. Здоровоохранение	3	1	2	Интерпретатор
Основы программирования. Команды ветвления и повторения	3	1	2	Фокала
Основы программирования. Числовые величины, машинная графика	5	1	4	KALK.FOC GRAF.FOC RIS.RIG FYNK.FOC
ЭВМ в современном обществе. Учебный процесс	11	1	10	GEOGR.RIG EVROPA.RIG BIOLOG.FOC ISTOR.RIG RUSBK.FOC ANGLBK.FOC LERM.RIG LIT.RIG FYNK.FOC FIZIK2.RIG FIZIK6.RIG BRIG.FOC PRAVO.RIG POST.RIG TELEK.RIG
Всего	36	6	30	

чатывать; при наборе цифры 1 на экране появляется перечень тем курса, при наборе цифры 2 печатается каталог нижней дискеты. Файлы каталогов содержат перечень используемых для раскрытия конкретных тем программ. На каждой дискете имеется один файл с именем CAT.CAT, который формируется на ДВК-2М в любом текстовом редакторе с этим именем. Выход из этого каталога — нажатием клавиши с точкой и ВК столько раз, сколько требуется для возврата в основной каталог. Выбираем здесь передачу в сеть. Набираем цифру 1. На вопрос ЭВМ,

какая программа (написанная на Фокале или в Т-редакторе) будет рассылаться по сети, следует набрать соответствующую цифру, затем имя передаваемого файла и номер БК-0010, на который передается файл.

Программы сгруппированы на дискетах по темам курса. Первая тема — знакомство с ЭВМ.

Первые 7 программ являются компьютерными играми. В методическом материале для преподавателя указываются педагогические цели отдельных занятий по темам курса; каждая используемая программа служит достижению этой це-

ли. Первое занятие дает достаточно времени для приобретения навыков работы отдельными, наиболее часто используемыми клавишами (управления курсором, «ввод», «пробел»); затем программа K.LAW.RIG позволяет освоить остальные клавиши. Программы KASOR.FOC, AUTO.RIG, SARAN.RIG, CHESS.RIG, OKEAN.RIG, K.LAW.RIG, ROBOT.RIG созданы в ИФТТ при ЛатвГУ им. П. Стучки (г. Рига) в системе «Рига»; LUNA.FOC разработана в горно-машиностроительном институте (г. Магнитогорск); LOGIC.FOC, REDAKT.RIG — в НИИИИВТ АПН СССР. Последняя программа дает представление о редакторе текстов и возможности использования ПЭВМ «Электроника БК-0010» для ввода и редактирования текстовой информации.

Программа ARHIV.RIG сделана в редакторе системы «Рига» и показывает, как можно использовать текстовый редактор при организации личного архива. ELTA64.FOC демонстрирует принцип работы электронных таблиц. Первая работает на монохромном дисплее, вторая рекомендуется для использования на цветном. BD64.FOC, BD32.FOC демонстрируют принцип работы информационно-поисковых систем. Первая программа рекомендуется при работе на монохромном дисплее, вторая — на цветном. LEGKIE.FOC, SERDS.FOC, TERPEN.FOC показывают возможности использования ЭВМ в здравоохранении. Они работают в диалоговом режиме и, проанализировав ответы пользователей, определяют степень риска: первая — легочных заболеваний, вторая — ишемической болезни сердца. Третья программа является психологическим тестом и определяет толерантность пользователя по отношению к окружающим.

Программа RIS.RIG, разработанная в ИФТТ при ЛатвГУ им. П. Стучки, знакомит с графическими возможностями ПЭВМ «Электроника БК-0010». При нажатии на определенные клавиши на экране возникают готовые рисунки (люди, дома, деревья и др.), komponуя которые можно создавать на экране дисплея целые картины. Программа GRAF.RIG (графический редактор) поз-

воляет рисовать на экране картины, состоящие из геометрических фигур, строить гистограммы, круговые диаграммы, делать к ним надписи. Построенные на экране диаграммы можно записывать на магнитную ленту. Программа FYNK.FOC позволяет рисовать на экране аналитически заданные функции (от одной до трех одновременно), выдает координаты указанных на графике точек, находит точки пересечения нарисованных графиков. Она также демонстрирует возможности применения ПЭВМ на уроках математики.

Пока БК-0010 поставляются с интерпретатором языка высокого уровня Фокал, поэтому 5-я и 6-я темы даются с его помощью. В методическом материале для преподавателя и слушателя описаны структуры Фокала, структура программ, реализация их на Фокале. Даны примеры их применения. KALK.FOC дает представление об использовании вспомогательных алгоритмов при решении сложных задач.

Учебные программы можно разделить на 4 вида: контролирующие (GEOGR.RIG, RUSBK2.FOC, LERM.RIG, ISTOR.RIG), тренажеры (ANGLBK.FOC) моделирующие (FIZIK2.RIG, FIZIK6.RIG), демонстрационные (EVROPA.RIG, BIOLOG.FOC, TRUD.FOC). GEOGR.RIG контролирует знание местонахождения городов СССР по карте, изображенной на экране БК-0010. EVROPA.RIG демонстрирует использование информационно-поисковой системы на уроке географии при изучении стран Европы (столица, население, занимаемая площадь). BIOLOG.FOC демонстрирует появление различных генотипов в потомстве при заданных генотипах родителей. RUSBK2.FOC контролирует знание правил правописания приставок «пре» и «при». ANGLBK.FOC — тренажер по словам английского языка при первом знакомстве с ним. ISTOR.RIG контролирует знание дат новой истории (IX класс средней школы), LERM.RIG контролирует знание дат из жизни и творчества М. Ю. Лермонтова. LIT.RIG — литературная викторина о творчестве русских писателей и поэтов. FIZIK2.RIG, FIZIK6.RIG моделируют

физико-химические явления. TRUD.FOC изображает на экране выкройку брюк по заданным размерам. BRIG.FOC, PRAVO.RIG, могут быть использованы в качестве деловых игр, т. е. для имитации принятия решения в различных производственных ситуациях. В первой программе идет расчет зарплаты каждому члену бригады рабочих в зависимости от коэффициента трудового участия. Задавая различные исходные данные, можно рассчитать зарплату по бригаде. Участник деловой игры принимает решения, опираясь на представленную ему информацию, как в начале, так и в процессе игры. Аналогично построена и вторая программа.

Стандарт на ПЭВМ

Нашим читателям, вероятно, будет интересно узнать, что с 1 июля 1987 г. действует впервые введенный ГОСТ 27201—87. Он определяет параметры, которым должны («Несоблюдение стандарта преследуется по закону!») соответствовать ПЭВМ, предназначенные для применения в социальной и производственной сферах пользователями-непрограммистами.

ГОСТ делит ПЭВМ на 5 типов (ПМ1—ПМ5). Машины типа ПМ1 рекомендуются для индивидуального применения в бытовых условиях, ПМ2 — для массового обучения, ПМ3 — для профессионального обучения и профессиональной деятельности (инженерных и экономических расчетов, планирования, работы с текстами), ПМ4 и ПМ5 — для профессиональной деятельности в научной, административно-управленческой, инженерной и других областях. ПМ1—ПМ3 могут быть 8- или 16-разрядными, ПМ4 — 16- или 32-разрядными, ПМ5 — не менее чем 32-разрядными. Минимальное быстродействие ПМ1 и ПМ2 — 500 тыс. коротких операций (типа регистр-регистр) в секунду, ПМ3 — 1 млн. оп./с, ПМ4 и ПМ5 — 2 млн. оп./с. Емкость ОЗУ ПМ1 и ПМ2 — не менее 64К байт (с 1991 г. емкость ОЗУ ПМ2 должна

В НИИИиВТ АПН СССР весной 1987 г. проводились занятия с преподавателями общеобразовательных предметов школ и системы профтехобразования Новосибирской области и Сибири, на которых использовался описанный пакет программ. Результаты опроса слушателей показали, что данный курс позволяет в короткий срок приобрести навыки работы на ПЭВМ «Электроника БК-0010», получить представление об использовании ЭВМ в некоторых областях деятельности нашего общества, способствует лучшему пониманию содержания курса ОИВТ, дает методику его изучения.

быть не менее 128К байт; далее в скобках будут приводиться значения параметра, вводимые с 1991 г.), ПМ3 — 512 К байт (1М байт), ПМ4 — 1М байт (2М байт), ПМ5 — 2М байт (8М байт). Впрочем, некоторые из этих величин могут быть уменьшены по требованию заказчика (потребителя), как и емкость накопителя на жестком диске, который является обязательной принадлежностью ПМ3—ПМ5 и в любом случае должен вмещать не менее 10М байт. Емкость же накопителя на гибком диске может меняться от 360К байт (для ПМ1) до 3М байт (для ПМ3—ПМ5 после 1990 г.). Для ПМ1 и ПМ2 по согласованию с заказчиком (потребителем) НГМД могут быть заменены на накопителя на кассетной магнитной ленте или магнитофоны. Прочтя это, те, кто хотел бы иметь дома хорошую ЭВМ с достаточным набором периферийных устройств, возможно, огорчатся; однако если бы ГОСТ требовал наличия в составе того же БК-0010 накопителя на гибких дисках, цена комплекта подскочила бы раза в три.

Для интересующихся компьютерной графикой: количество адресуемых точек экрана устанавливается в основном равным 640×200, а для ПМ5 и, с 1991 г., для ПМ4 — 720×512; цветной

монитор обязателен только для ПМ5.

Обязательная часть ПЭВМ — базовый комплект — должна включать микропроцессор, ОЗУ, клавиатуру, устройство отображения информации (в базовом комплекте ПМ1 его может не быть, если предполагается подключение к бытовому телевизору), адаптеры и контроллеры для подключения периферийных устройств. Но самое, пожалуй, важное, что в базовый комплект включены не только операционная система, но и пакеты прикладных программ общего назначения. Состав этих пакетов, как и многие другие особенности ПЭВМ, могут, по ГОСТу, задаваться в ТЗ и ТУ, однако системы программирования, функционально-ориентированные пакеты прикладных программ общего назначения, проблемно-ориентированные па-

кеты прикладных программ и тестовые программы (проверяющие работоспособность ПЭВМ) должны быть обязательно. Программное обеспечение ПМ3—ПМ5 должно включать пакеты, обеспечивающие обработку текстов, графической информации, управление базами данных, трансляцию с Бейсика, Фортрана, Паскаля, Си, работу в локальной сети, «электронные таблицы».

И последнее (по счету, но не по существу дела) — надежность. Базовый комплект ПЭВМ должен обеспечивать наработку на отказ не менее 10 тыс. ч, а с 1991 г. не менее 15 тыс.ч; среднее время восстановления работоспособности ПЭВМ должно быть не более 30 мин, а с 1991 г.— не более 15 мин. Нарботка на отказ КУВТ должна быть не менее 5 тыс. ч., с 1991 г.— 10 тыс. ч.

65

В ПОМОЩЬ УЧИТЕЛЮ

Печатные учебно-наглядные пособия для уроков информатики

С 1986 г. издательство «Просвещение» выпускает демонстрационные печатные пособия по курсу ОИВТ. Первой вышла серия таблиц для IX класса «Алгоритмы и алгоритмический язык. Алгоритмы работы с величинами» Э. И. Кузнецова и И. В. Роберт (тираж — 60 тыс. экз.). В нее вошли таблицы: понятие алгоритма; служебные слова алгоритмического языка; составные команды алгоритмического языка; способы записи алгоритма; исполнение алгоритма; алгоритмы работы с графической информацией; основные библиотечные алгоритмы. Первые шесть иллюстрируют теоретические основы курса информатики, остальные посвящены основным библиотечным алгоритмам, изучаемым в IX классе. Использование пособия зависит от форм, методов и целей урока. Его можно применять при объяснении нового материала, проведении самостоятельных работ, в процессе опроса, закрепления и повторения, как справочный материал. Часть таблиц рекомендуется постоянно экспонировать в кабинете информатики.

В методическом руководстве, прилагаемом к таблицам, даны рекомендации по работе с ними, предложены варианты их использования.

В 1987 г. выпущены две серии таблиц. Первая — триптих «Перспективы развития вычислительной техники в свете решений XXVII съезда КПСС» (автор — В. П. Долматов; тираж — 60 тыс. экз.), вторая — «Структура и принципы работы ЭВМ» (авторы — Г. М. Нурмухамедов и А. А. Кузнецов; тираж — 87 тыс. экз.). Она предназначена для X класса и содержит следующие таблицы: структурная схема ЭВМ; структура микро-ЭВМ; одноплатная микро-ЭВМ «Электроника МС 1201»; структура центрального микропроцессора; команды микро-ЭВМ; некоторые команды микропроцессора K1801BM1; программный принцип работы ЭВМ; пример машинной программы; основные виды памяти микро-ЭВМ; оперативное запоминающее устройство — ОЗУ; накопитель на гибком магнитном диске; устройства ввода и вывода

информации; поколения ЭВМ. Содержание некоторых таблиц выходит за рамки программы общеобразовательной школы, пособие можно использовать в школах с углубленным изучением физико-математических дисциплин, на факультативных занятиях, в СПТУ и средних специальных учебных заведениях.

Комплект включает объяснительный текст, который не дублирует учебное пособие и должен помочь учителю расширить свои знания о принципах работы ЭВМ. В нем приведены также список аббревиатур, принятых в информатике, с их расшифровкой, перечень дополнительной литературы.

В 1988 г. планируется издать два пособия по языкам программирования Бейсик и Рапира.

Демонстрационное пособие «Язык программирования. Бейсик» А. С. Лесневского включает 11 таблиц: общий вид программы на Бейсике; постоянные, переменные величины; запись табличных величин; операции; стандартные функции; команда присваивания; арифметические выражения; команда ветвления; команда повторения; команда заполнения подпрограммы; команда выбора; команды ввода и вывода информации; графические команды. Предполагаемый тираж — 50 тыс. экз.

В пособии рассмотрены основные синтаксические конструкции этого языка, приведены примеры программ, записанных на Бейсике. Большинство таблиц серии посвящено правилам записи команд.

Таблицы сопровождаются пояснительным текстом, в котором приводятся дополнительные теоретические сведения, необходимые для работы с иллюстративным материалом, рассматриваются примеры, не вошедшие в наглядный ряд. Для учителя предназначены пояснения механизма действия отдельных команд, ряд программ, записанных на Бейсике, транскрипция и перевод основных слов языка.

Пособие «Язык программирования. Рапира» Ю. А. Первина состоит из 17 таблиц: синтаксические диаграммы; переменные величины, их имена и значения; арифметика Рапиры; команда вывода; тексты (литерные

величины); кортежи; операции над кортежами; условия (логические значения); команды ветвления; циклы (цикл ПОКА, цикл ДЛЯ, вложенные циклы); команда ввода; процедуры; описание и вызов процедур; процедуры с параметрами; процедуры вычисления функций; стандартные функции Рапиры. Предположительный тираж пособия — 50 тыс. экз.

Для знакомства с языком в таблицах выбран метод синтаксических диаграмм, дающий возможность в наглядной и доступной форме показать учащимся основные правила построения программ в Рапире.

В объяснительном тексте автор приводит и уже знакомые ученикам, и новые понятия, необходимые для работы с таблицами, давая определение каждому из них. Сюда же включены примеры, которые предлагается разобрать на уроке. Автор дает рекомендации по использованию таблиц на уроке, помогая учителю выделить основные вопросы при объяснении материала.

Обе серии таблиц по языкам программирования являются дополнением к действующему пробному учебному пособию по основам информатики и вычислительной техники. Однако содержание таблиц несколько шире школьной программы, что позволяет использовать их для углубленного изучения языков на факультативных занятиях и в специализированных учебных заведениях.

Описанные учебные пособия можно приобрести в специализированных магазинах Главснабпроса РСФСР. Заказы на них следует направлять в адрес магазина, к которому школа прикреплена.

Сообщая об издании таблиц, издательство «Просвещение» надеется на отклик учителей и их помощь в деле совершенствования учебно-наглядных пособий по информатике и вычислительной технике. Свои отзывы, замечания и предложения просим направлять по адресу: 129846, Москва, 3-й проезд Марьиной рощи, 41, редакция наглядных пособий.

Н. ТЮРИНА

Редактор редакции наглядных пособий
издательства «Просвещение»

К сведению читателей

Создано новое издательство «КНИЖНАЯ ПАЛАТА».

**«Компьютер в библиотеке» — одна из серий,
которая начнет выходить в ближайшее время.**

**Первая книга этой серии будет называться
«Персональные ЭВМ в библиотеке».**

Г. ГОРОДИЛОВА, П. САМОРОДНИЦКАЯ, Р. АЛЛАХВЕРДИЕВ

Компьютеры и орфография

Как известно, понятное и точное предписание исполнителю о совершении элементарных операций в логически целесообразной последовательности для решения любой из задач, принадлежащих данному классу, называется алгоритмом.

Пошаговое разбиение дидактического материала, соответствующее мыслительным процессам обучаемого и представление его в виде блок-схемы составляет содержание первой части статьи.

Во второй части излагаются приемы реализации конкретного алгоритма в учебном процессе педагогического вуза с помощью компьютерной техники при подготовке преподавателей русского языка.

Подвергнув анализу орфографические минимумы, представленные в различных учебных пособиях по русскому языку, мы пришли к выводу, что множество частных задач может быть минимизировано путем сведения их к более общей задаче, решаемой на основе единого алгоритмического предписания.

Проиллюстрируем сказанное на примере тем по орфографии в практическом курсе русского языка.

Написание *н* и *нн* в суффиксах различных частей речи рассматривается при изучении полных и кратких прилагательных, полных и кратких причастий и мотивированных ими наречий на *-о* и существительных на *-ник* (*-ница*). Как известно, различение *н* и *нн* вызывает у учащихся серьезные затруднения и приводит к наиболее частотным орфографическим ошибкам.

Алгоритмическое описание названного правила позволяет построить алгоритм решения этой задачи независимо от того, какой частью речи являются слова с *н* и *нн*

в суффиксах: для прилагательных и причастий это обусловлено тем, от какой части речи — существительного или глагола — они образованы, для наречий на *-о* и существительных на *-ник*, (*-ница*) — от чего образовано мотивирующее их слово.

Ниже приводим алгоритмическое описание рассматриваемого правила.

Если интересующее нас слово образовано от существительного, то оно пишется с одним *н* при следующих условиях:

если оно образовано с помощью суффиксов *-ан-*, *-ян-*, *-ин-*;

если оно не входит в список исключений (*стеклянный*, *оловянный*, *деревянный*).

В противном случае — если оно образовано с помощью *-енн-*, *-онн-* или с помощью суффикса *-н-* от слова с основой на *н* и если это не слово-исключение (*ветренный*), оно пишется с двумя *нн*.

Если интересующее нас слово образовано от глагола, то оно пишется с одним *н* при следующих условиях:

если мотивирующий глагол несовершенного вида;

если в слове нет суффикса *-ова-* (*-ева-*);

если при нем нет зависимого слова;

если оно не входит в число исключений (*неслыханный*, *невиданный*, *нежданный*, *негаданный*, *желанный*, *священный*, *нечаянный*, *медленный*, *деланный*);

если это слово-исключение (*приданое*);

если оно является кратким страдательным причастием.

В противном случае, если мотивирующий глагол совершенного вида; если слово имеет суффикс *-ова-* (*-ева-*); если имеется зависимое слово; если оно входит в первый список исключений и отсутствует во втором, оно пишется с *нн*.

Как явствует из алгоритмического описания, предписание может быть сведено к шести элементарным операциям по определению, пяти по проверке принадлежности к множеству исключений (и вытекающим из каждой операции логическим условиям) и двум логическим заключениям.

Операции по определению:

1. Образовано ли слово от существительного.

2. Образовано ли слово от существительного с основой на н с помощью суффикса -н-.

3. Образовано ли слово от существительного с помощью суффиксов -енн-, -онн-.

4. Образовано ли слово от глагола совершенного вида.

5. Имеется ли в слове суффикс -ова-, (-ева-).

6. Имеется ли при нем зависимое слово.

Операции по проверке принадлежности к исключению или к кратким страдательным причастиям:

1. Входит ли в число исключений (ветренный).

2. Входит ли в число исключений (оловянный, деревянный, стеклянный).

3. Входит ли в число исключений (приданое).

4. Входит ли в число исключений (незданный, негаданный, невиданный, неслыханный, желанный, священный, нечаянный, медленный, деланный).

5. Является ли кратким причастием.

Логические заключения:

Пиши **нн**.

Пиши **н**.

Мы считаем рациональнее делать объектом выработки навыка орфограмму, представляющую собой некоторую условность для норм орфографии.

Так, если в суффиксах может писаться как **нн**, так и **н**, объектом запоминания и заучивания должна быть орфограмма **нн**, так как написание **н** естественнее, ежели **нн**. Поэтому алгоритм нацелен на те операции, которые приводят к написанию **нн** (образовано ли слово от основы на н с помощью суффикса -н-, с помощью суффиксов -енн-, -онн-, от глаголов совершенного вида, от глаголов на -ова- (-ева-), имеет ли себе зависимое слово, входит в блоки исключений).

Если слово не удовлетворяет этим логическим условиям, то в нем нет **нн**, т. е. оно пишется с **н**.

Операция по проверке принадлежности к множеству исключений всегда связана с ограниченным числом случаев, поэтому с точки зрения учебной задачи эти операции непосредственно предшествуют логическому заключению.

Блок-схема алгоритма представлена в табл. 1.

Написание **о**, **ё** после шипящих под ударением рассматривается при изучении различных тем: состав слова; окончания существительных; окончания и суффиксы прилагательных; наречия на -о; суффиксы причастий, отглагольных прилагательных и существительных; личные окончания глаголов. Неразличение **о**, **ё** после шипящих также вызывает частотные орфографические ошибки.

Правописание **о**, **ё** после шипящих под ударением зависит от того, в какой части слова находится интересующий нас гласный. Если в корне слова, правописание его зависит от наличия или отсутствия в другой форме слова или в родственном слове чередования интересующего нас гласного с **е**; при наличии чередования с **е** пишется **ё**, при отсутствии — пишется **о**. Если в суффиксе или окончании слова, правописание интересующего нас гласного зависит от того, какая это часть речи: в существительных, прилагательных, наречиях под ударением пишется **о** (суффиксы -ок-, -онк-, -онок-, -ов-, -о-; окончания -о, -ой, -ом, -ого); в глаголах, причастиях, отглагольных существительных и прилагательных под ударением пишется **ё** (суффиксы -ённ-, -ён-, ён+к-, -ёвк-; личные окончания глаголов).

Как явствует из алгоритмического описания, предписание может быть сведено к трем элементарным операциям, по определению (с возникающими из каждой положительными и отрицательными условиями):

1. Находится ли интересующий нас гласный в корне.

2. Имеется ли переход интересующего нас гласного в **е**.

3. Являются ли слова с интересующим нас гласным существительными (суффиксы -ок-, -онк-, -онок-, окончания -о, -ом, -ой), прилагательными (суффикс -ов-) или наречиями (суффикс -о).

Логические заключения:

Пиши **ё**.

Пиши **о**.

Блок-схема алгоритма представлена в табл. 2.

Написание **ь** на конце слов после шипящих изучается в следующих темах программы: правописание родительного падежа множественного числа на шипящий; употребление **ь** на конце существительных после шипящих; краткие прилагательные мужского рода с основой на шипящий; мягкий знак после шипящих в глаголах неопределенной формы; в форме II лица единствен-

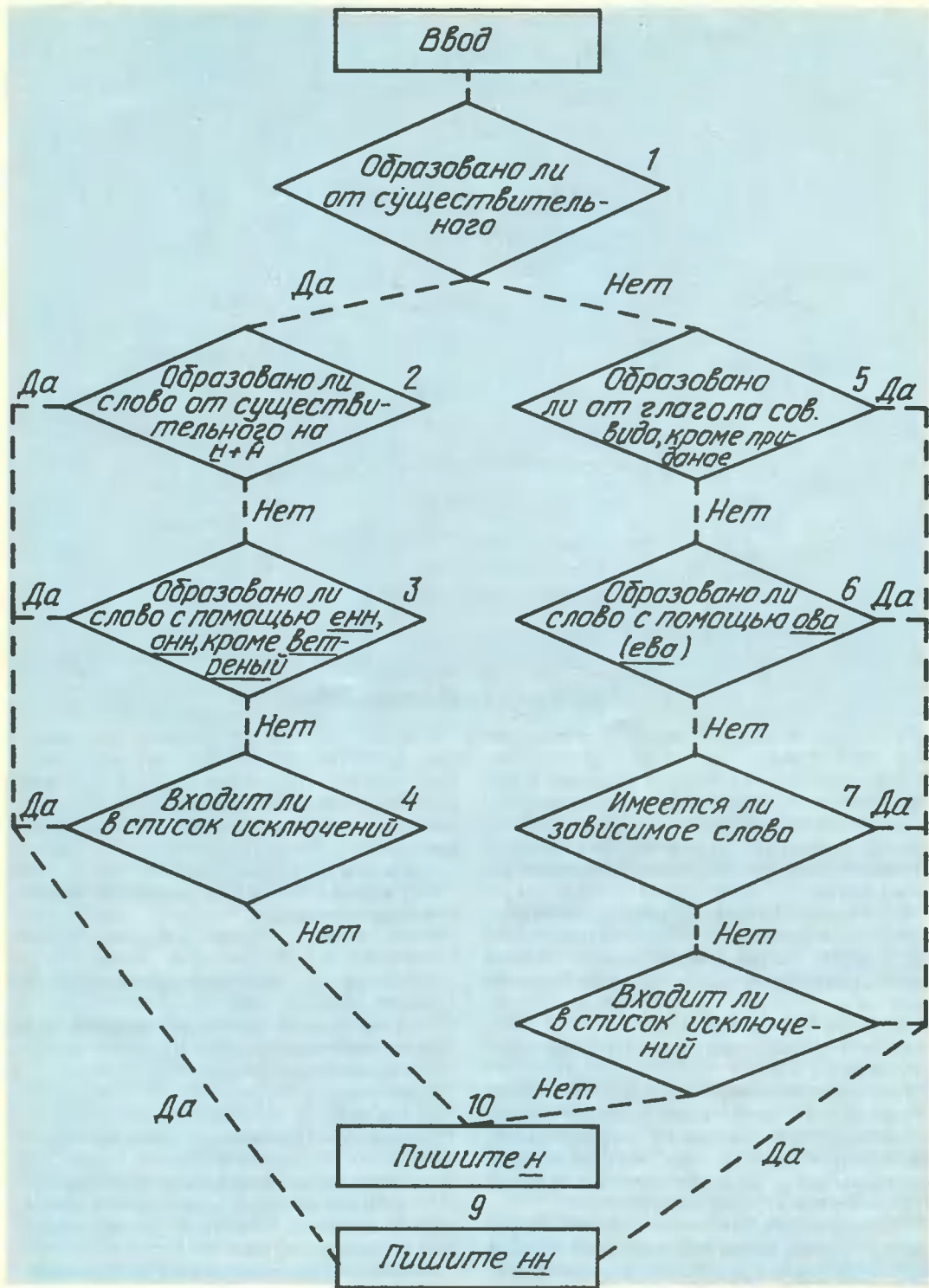


Схема алгоритма **н** или **нн** в суффиксах различных частей речи

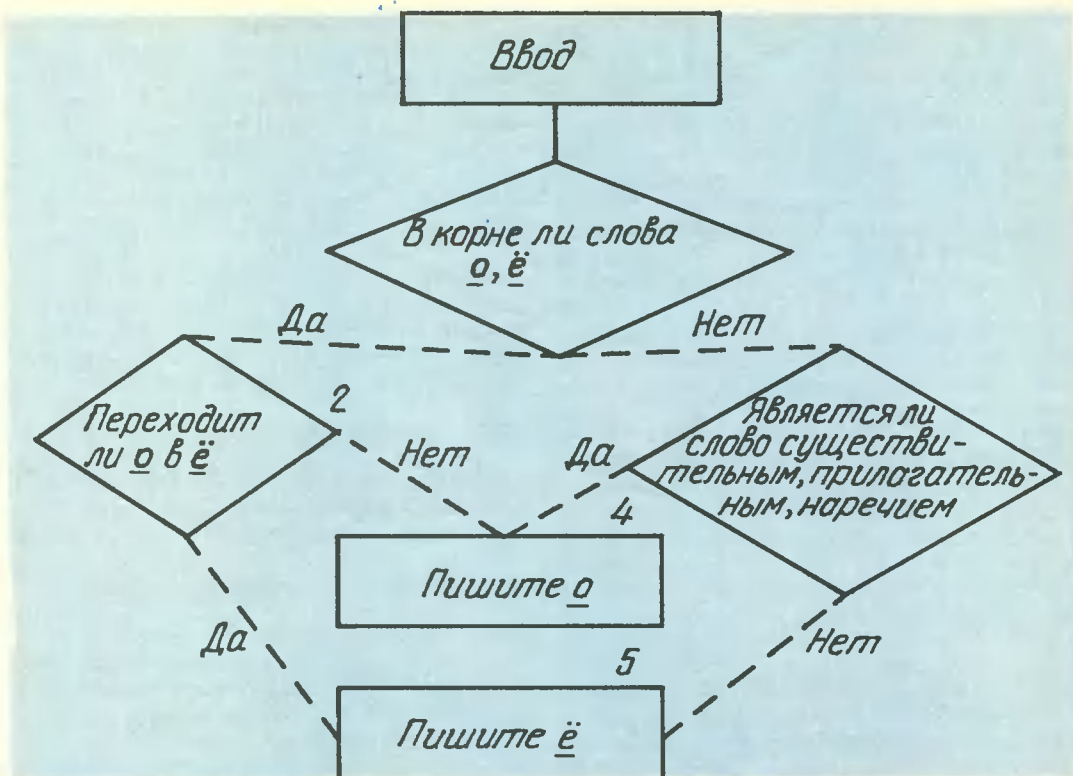


Схема алгоритма О, Ё после шипящих

ного числа; на конце глаголов повелительного наклонения.

Объединение всех задач, связанных с его написанием, в общий метод решения позволяет свести множество задач к трехшаговому алгоритму. Дадим алгоритмическое описание правила о ь после шипящих на конце слова.

Если интересное нас слово, оканчивающееся на шипящий, — существительное мужского рода, форма родительного падежа множественного числа с нулевым окончанием или краткое прилагательное мужского рода, оно пишется без ь.

Если это существительно III склонения, оно пишется с ь.

Если интересное нас слово — глагол (в форме II лица единственного числа, в неопределенной форме, в повелительном наклонении), наречие или частица и если оно не входит в число исключений (уж, замуж, невтерпеж), оно пишется с ь.

Объектом изучения и запоминания следует считать случаи правописания слов с ь после шипящих, так как отсутствие ь после шипящих — явление естественное, поскольку никакой фонетической функции ь не выполняет.

Как явствует из алгоритмического описания, алгоритм состоит из двух элементарных операций по определению, одной операции по проверке принадлежности к множеству исключений и двух логических заключений.

Операция по определению:

1. Является ли слово существительным или прилагательным.
2. Является ли слово существительным III склонения.

Операция по проверке принадлежности к исключениям:

1. Входит ли в состав исключений (уж, замуж, невтерпеж).

Логические заключения:

1. Пиши ь.
2. Не пиши ь.

Блок-схема алгоритма представлена в табл. 3.

Рассмотрим приемы реализации алгоритма в учебном процессе с помощью компьютерной техники, предоставляющей следующие возможности.

Компьютер представляет на дисплее алгоритм в приближенном к наглядному пособию виде; задания; текстовый материал для выполнения задания.

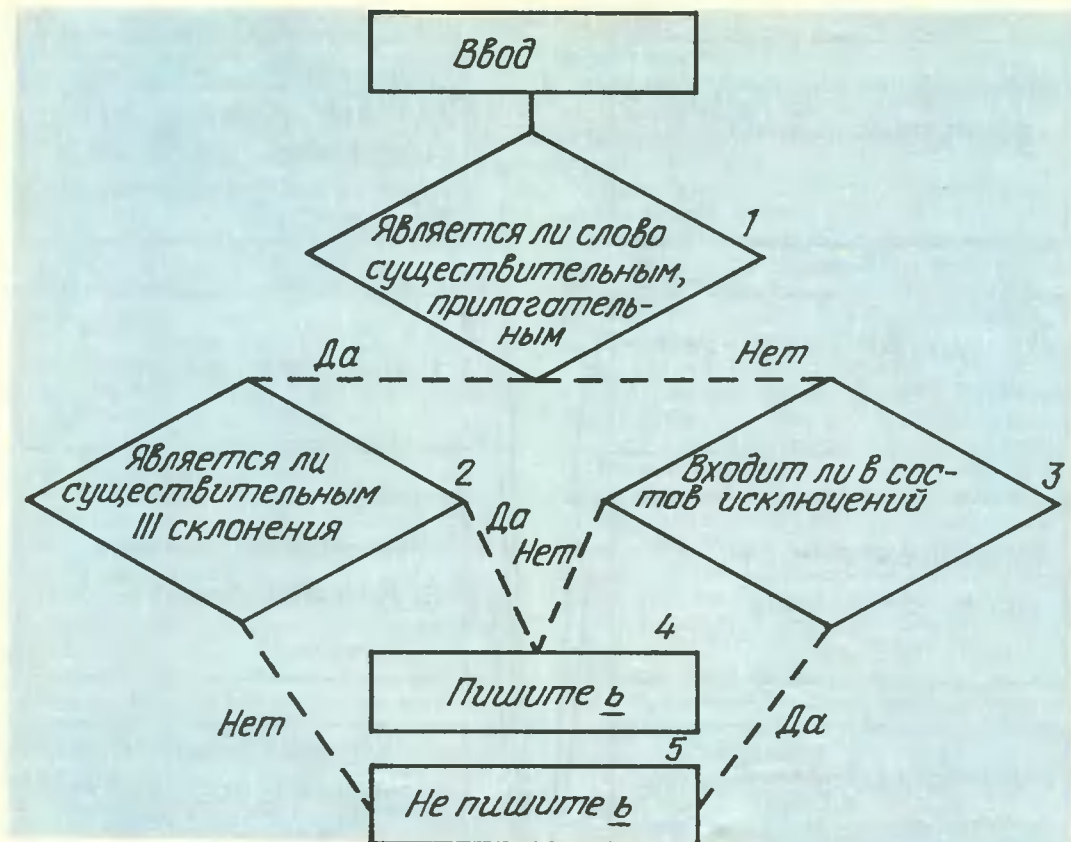


Схема алгоритма Б после шипящих на конце слов

Ответы студентов вводятся в компьютер набором букв, подчеркиванием, набором номеров блоков алгоритма или в принятом коде — «да» или «нет».

Если написание той или иной орфограммы связано с другим языковым материалом, от владения которым зависит усвоение изучаемой орфограммы, компьютер может предъявить добавочную порцию материала (подпрограмму). Если на такое правило допущена ошибка, компьютер «исправляет» ошибку, направляя студента к блоку алгоритма, предписывающему написание данной орфограммы, и предъявляет то же задание на другом дидактическом материале. При повторной ошибке компьютер отправляет студента к подпрограмме, соответственно снабженной кроме добавочного материала своими заданиями. Только после того как студент справился с ними, компьютер возвращает его к основной программе.

Если ошибка допущена в случае, не нуждающемся в обращении к подпрограмме, она (ошибка) «исправляется» компьютером описанным выше способом.

Только после правильного выполнения задания студент получает следующую порцию учебного материала.

Задание, в котором была допущена ошибка, отсроченно повторяется.

Компьютеру задается критерий усвоения. Он складывается из отношения числа неправильных ответов к общему числу ответов; относительного веса ошибки; времени, затраченного на ответ.

По выполнении каждого упражнения компьютер проставляет студенту индивидуальный балл, слагаемый из перечисленных выше параметров. Если индивидуальный балл не удовлетворяет принятому критерию усвоения, студент получает дополнительное упражнение на другом дидактическом материале.

Использование компьютерной техники рассматривается далее в процессе поэтапной выработки орфографических навыков.

Согласно концепции П. Я. Гальперина, поэтапное формирование умственных действий проходит этапы внешнего действия, имеющего ориентировочную и исполнитель-

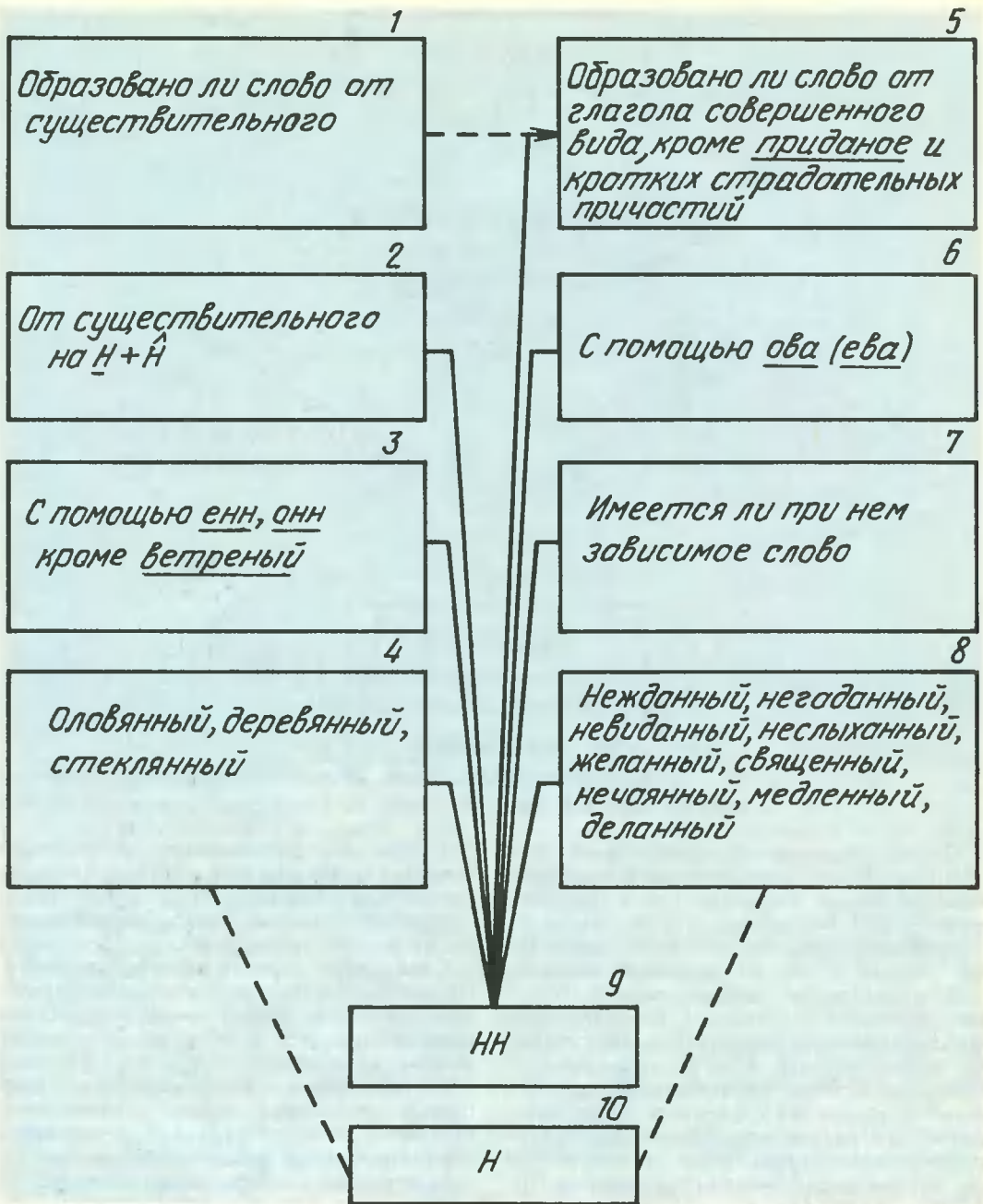


Схема алгоритма Н или НН в суффиксах различных частей речи

ную части, и внутреннего действия. В процессе формирования действие должно быть развернуто, обобщено и свернуто, что и составляет основу поэтапного формирования умственного действия.

Как известно, для формирования навыка необходимо перевести ориентировочную

часть действия во внутренний план, в результате чего внешнее рабочее действие станет автоматическим процессом, исполнение которого не требует осознания, что и характеризует сформированность навыка.

Эта теория легла в основу процесса поэтапной выработки орфографических на-

выков, включающего следующие этапы и их составляющие:

1. Ориентировочный этап.

1.1. Ознакомление с полной системой ориентиров (наблюдение).

1.2. Уяснение системы ориентиров (при выполнении аналитических видов работ на соотношение, группирование, подбор дидактического материала).

2. Исполнительный этап.

2.1. Действие с полнотой охвата системы ориентиров.

2.1.1. Отработка действия с опорой на материализованный образец.

2.1.2. Отработка действия без опоры на материализованный образец.

2.2. Действия с постепенным свертыванием системы ориентиров.

3. Этап осуществления действия, ориентировочная часть которого перешла во внутренний план, что свидетельствует о сформированности навыка.

Процессу выработки орфографических навыков на ориентировочном и исполнительном этапах соответствуют языковые упражнения, в то время как на этапе сформированного внутреннего действия операция включается в речевое действие, чему соответствуют речевые упражнения (процесс выработки умений).

При выработке орфографического навыка с помощью компьютерной техники могут быть реализованы языковые упражнения.

Далее предлагается типология заданий для компьютерной техники, иллюстрируемая алгоритмом на различение *н* и *нн* в суффиксах.

Пошаговое предъявление алгоритма в виде наглядного материала (представлено в табл. 4).

Поскольку на первом этапе все студенты еще работают одновременно, предъявление алгоритма осуществляется синхронно на всех дисплеях, что делает возможным сопровождение его словом преподавателя, иллюстрирующим примерами и объясняющим каждый шаг алгоритма.

Иллюстрация алгоритма примерами.

Предлагается по одному слову с изучаемой орфограммой (она выделена в слове вместе с суффиксом), иллюстрирующему последовательно каждый блок алгоритма.

Предъявление каждого слова сопровождается дополнительным высвечиванием соответствующего блока алгоритма.

1.2. Предъявление вопросов, соответствующих каждому блоку алгоритма.

Задания состоят из вопроса и списка слов (для ответа). Ответ «да» или «нет» вводится в принятом коде.

Образовано ли слово от существительного?

Образовано ли слово от существительного на *-н*?

Образовано ли слово от существительного с помощью суффиксов *-енн-*, *-онн-*?

Образовано ли слово от глагола совершенного вида?

Образовано ли слово от глагола с суффиксами *-ова-*, *-ева-*?

Имеется ли к подчеркнутому слову зависимое слово (в дидактическом материале для данного задания, в отличие от других, интересующие нас слова предложены в контексте)?

Входит ли слово в список исключений?

При подборе дидактического материала включаются и слова с «ложными» ориентирами (образованные от бесприставочных глаголов совершенного вида и от приставочных глаголов несовершенного вида с *ов*, *ев*, не являющимися глагольными суффиксами; с *ин*, *он*, *ен* в основе мотивирующего существительного; слова, имеющие внешнее сходство с исключениями).

2.1. Задание на указание студентами пути следования по алгоритму для каждого из предъявленных списком слов.

Ответ вводится набором блоков алгоритма с указанием «да» или «нет» по каждому из проходимых блоков.

Так, для слова **лимонный** следует указать блоки 1 «да», 2 «да», 9;

для **соломенный**, **революционный** — 1 «да», 2 «нет», 3 «да», 9;

для **ветреный** — 1 «да», 2 «нет», 3 «да», 10;

для **деревянный** — 1 «да», 2 «нет», 3 «нет», 4 «да», 9;

для **кожаный** — 1 «да», 2 «нет», 3 «нет», 4 «нет», 10;

для **переваренный** — 1 «нет», 5 «да», 9;

для **приданое** — 1 «нет», 5 «да», 10;

для **рисованный** — 1 «нет», 5 «нет», 6 «нет», 7 «да», 9;

для **нечаянный** — 1 «нет», 5 «нет», 6 «нет», 7 «нет», 8 «да», 9;

для **вареный** — 1 «нет», 5 «нет», 6 «нет», 7 «нет», 8 «нет», 10.

Задание выполняется в двух вариантах (на разном дидактическом материале).

В первом варианте студент получает пошагово информацию о правильности выполнения действия; только при правильном указании им номера очередного блока и кода ответа нужный блок дополнительно высвечивается.

Во втором варианте задания, аналогичном предыдущему, информацию о правильности выполнения действия студент получает только по окончании работы по алгоритму; в случае ошибки указывается блок, в котором ошибка допущена.

Задание на словообразование с *н* и *нн*

в суффиксах. Предъявляются мотивирующие слова с подчеркнутой производящей основой и весь набор словообразующих суффиксов: -енн-, -онн-, -н-, -ан-, -ян-, -ин-, -яин-, -ен-, -нн-; задание выполняется с опорой на материализованный образец (алгоритм, предъявляемый на дисплее). Дидактический материал, предъявляемый для выполнения задания, также составляет материализованную основу. Выполнение осуществляется набором букв и указанием пути следования по алгоритму (как в задании 2.1).

2.1.2. Задание на составление по памяти усвоенного алгоритма.

Предлагаются блоки (без их номеров) в произвольной последовательности и контуры алгоритма с номерами блоков; выполнение задания осуществляется указанием номера для каждого блока. При правильном указании студентам номера блока он появляется в своем контуре; отсутствие его на дисплее является сигналом об ошибке и необходимости ее исправить. И так до полного составления всего алгоритма.

Если выполнение задания не удовлетворяет критерию усвоения, предлагается выполнить его заново. Для повторного выполнения блоки даются в другой последовательности.

Задание на вписывание н или нн в данные список слова с многоточием на месте пропущенной орфограммы с указанием (как и в предыдущем задании) пути следования по алгоритму. Последним обеспечивается полнота охвата ориентиров и снимается вероятность случайности правильного решения задачи.

Вписывание орфограммы осуществляется набором букв; путь следования по алгоритму — набором номеров блоков алгоритма и вводом «да» или «нет» в принятом коде.

2.1.3. Задание на вставку орфограммы н или нн без материализованной опоры на алгоритм, а только с указанием пути следования операций, что обуславливает полноту охвата ориентиров. Вставка н и нн осуществляется набором букв, указание пути следования по алгоритму осуществляется, как в задании 2.1.

Задание на словообразование прилагательных (в контексте).

Задание предлагает заменить словосочетания со связью управления словосочетаниями со связью согласования, чем осуществляется словообразование прилагательного от существительного. Словообразование

предполагает использование того или иного суффикса с изучаемой орфограммой. Ответ сводится набором букв и указанием номеров блоков алгоритма, чем осуществляется полнота охвата ориентиров. Ошибки, не связанные с изучаемой орфограммой, только исправляются компьютером — миганием орфограммы, в которой допущена ошибка, и заменой ее правильной, которая также мигает, чтобы быть замеченной.

2.2. Задание на словообразование. Задание на образование притягательных (со свертыванием системы ориентиров). Список существительных для образования прилагательных предполагает использование суффиксов блока 2, затем блоков 2—3, затем блоков 2—3—4, а также суффиксов с н, чем предусматривается свертывание системы ориентиров.

Задание на образование от глаголов слов с изучаемой орфограммой с выполнением следующих дополнительных заданий: указать вид глагола, подчеркнуть производящую основу, выделить словообразующий суффикс.

Задание на образование слов от глаголов с выделением в глагольной основе суффиксов -ова- (-ева-) (если он имеется).

Задания на вставку н и нн.

Задание на вставку н или нн с указанием зависимого слова, если оно имеется (слова для вставки даются в контексте).

Задание на вставку н или нн (в словах для вставки перемежаются исключения и слова, внешне напоминающие исключения).

Задание на вставку н и нн с распределением слов по соответствующим столбикам.

Задания на перестройку текста.

Задание на трансформацию определительных словосочетаний в предикативные с заменой полных прилагательных краткими (дидактический материал предполагает использование суффиксов блока 2, затем блоков 2—3).

Задание на перестройку текста с вставкой зависимого слова (что сопровождается изменением в написании н и нн).

Задание на перестройку с заменой одного из однородных глагольных сказуемых общими определениями.

Задание на трансформацию текста с образованием существительных на -ник (-ница).

Задания изложенного этапа направлены на постепенное свертывание системы ориентиров. В случае ошибки студент возвращается к тому же заданию, но с указанием всей системы ориентиров (с указанием пути следования по алгоритму).

Проблемы компьютерного обучения в вузе

Важность компьютерного обучения студентов подчеркивалась и в «Основных направлениях перестройки высшего и среднего специального образования в стране». Отмечалось, что в двенадцатой пятилетке необходимо обеспечить потребность учебного процесса в электронно-вычислительной технике. Предстоит создать 130 тысяч рабочих мест, оборудованных персональными компьютерами, терминальными устройствами.

С какими проблемами сталкивается вуз при широком внедрении ЭВМ? Что мешает эффективно использовать их в институте? Об этом рассказывается в этой статье.

В Московском инженерно-физическом институте применяется разнообразная вычислительная техника. Здесь установлены пять ЭВМ типа ЕС, с помощью которых ученые института и студенты старших курсов выполняют сложные расчеты, требующие большого объема вычислений. Есть на кафедрах и мини-ЭВМ типа СМ, применяемые для обработки результатов научных исследований. В нашем институте действует совет по вычислительной технике, который активно помогает ее внедрению в учебный процесс и научно-исследовательскую работу. Специалисты МИФИ совершенствуют имеющуюся материально-техническую базу, разрабатывают программное обеспечение, по мере сил занимаются обслуживанием и ремонтом вычислительной техники. Например, в нашем институте создан унифицированный микропроцессорный учебный лабораторный комплекс для автоматизации лабораторных работ и практикумов. Используя вычислительную технику в управлении лабораторным оборудованием, физической аппаратурой, мы ввели новый комплекс обучения студентов.

Вот некоторые его особенности:

проблемная ориентация вводимых технических и программных средств;

лучшая реализация измерительно-управляющих функций;

стандартный подход к разработке узлов и блоков;

применение унифицированных модулей.

В основе комплекса — микропроцессор, видеотерминал, набор интерфейсных блоков для подключения физической аппаратуры, устройства ввода—вывода. Отображение графической информации на экране дисплея

осуществляется с помощью растровой графической системы ГРАД-МИКРО, также разработанной в МИФИ.

В программном обеспечении комплекса есть пакет системных программ, программные модули формирования графических структур, интерпретатор языка Бейсик для описания сценариев проведения лабораторных практикумов.

Пакет графических модулей представлен в виде пополняемой библиотеки процедур, управляемой внутренним монитором. Практика показала, что применение такой системы для проведения лабораторных работ повышает эффективность обучения студентов.

К сожалению, его компьютеризацию мы проводим в основном своими силами. А в этом деле так необходима помощь предприятий и организаций, выпускающих и обслуживающих вычислительную технику. Например, нам не хватает персональных ЭВМ, на которых студенты могли бы работать с первого курса. Если бы первокурсники сразу же включились в диалог с компьютером, эффективность подготовки будущих специалистов значительно бы возросла. Но такую систему обучения создать очень трудно. Ведь отечественные персональные ЭВМ «Искра», ДВК и другие, которые используются сейчас в вузе, ненадежны, а их стандартное математическое обеспечение не на высоте.

Пришла пора организовать широкое тиражирование учебных компьютерных программ. Тогда студент на занятия вместе с конспектами приносил бы персональный диск с программой обучения по тому или иному предмету. Имея такой диск, студент мог бы работать с компьютером по строго индивидуальной программе и быстрее усваивал материал. При этом число студентов, получивших навыки самостоятельной работы с ЭВМ, значительно бы возросло.

Но пока дисков не хватает, а тот минимум, который поставляется с отечественными персональными ЭВМ,— весьма низкого качества.

Чтобы в создавшихся условиях нехватки учебных персональных компьютеров приобщить как можно больше студентов к активному диалогу с ЭВМ, в нашем институте разработаны вычислительные сети коллективного пользования. В них к одной ЭВМ под-

ключается несколько видеотерминалов. Такая сеть успешно действует на нескольких кафедрах института, способствуя более широкому доступу студентов к вычислительной технике.

Необходимо, чтобы информационные сети кафедр объединились в общеинститутскую, а в перспективе в междузубовскую сеть. Об этом говорится и в «Основных направлениях перестройки высшего и среднего специального образования в стране». Подчеркивается, что необходимо расширить работы по организации междузубовских сетей коллективного пользования, банков данных, улучшить информационное обслуживание процесса обучения и научных исследований.

76

Повысить эффективность работы вычислительных сетей можно с помощью оптоволоконных систем связи. Сейчас специалисты нашего института разрабатывают такую систему совместно с Министерством электронной промышленности, которое поможет нам в поставках оптического волокна, в то время как наш институт разработал для нужд министерства оптические разветвления. Конечно, спроектировать информационную сеть можно силами института, но где ему достать необходимую аппаратуру связи, видеотерминалы, персональные ЭВМ? Их отсутствие приводит к тому, что интересные зубовские начинания порою так и остаются на бумаге. Но предположим, что институту удалось достать необходимую вычислительную технику. Кто же должен заниматься ее обслуживанием и ремонтом? По идее, те предприятия, которые ее выпускают. Ведь что получается? Отправит зубу то или иное предприятие компьютер, и, как говорится, с плеч долой. Пусть институт сам заботится об этом. Но ведь он не выпускает запасных частей и не имеет штата наладчиков. Например, в нашем институте сейчас используются персональные компьютеры «Искра-22б», выпускаемые одним из приборостроительных заводов. На все наши просьбы помочь в обслуживании этих машин заводчане отвечают отказом. Мы пытались организовать ремонт «Искры» своими силами. Но для этого нужны запасные детали, поставлять которые завод отказывается. И если наш институт еще кое-как способен решать проблему обслуживания компьютером, что делать гуманитарным зубам, где нет специалистов по вычислительной технике? Крайне трудно в зубе обслуживать имеющуюся вычислительную технику. Не хватает специально оборудованных помещений, нет рекомендаций по оптимальному размещению персональных компьютеров в аудиториях и лабораториях институтов. При существующей системе обслуживания персо-

нальных ЭВМ каждый зуб должен создавать у себя своеобразный вычислительный центр со штатом наладчиков, программистов, операторов. Это приводит к расходам, которые многим зубам не по карману. Надо организовать централизованное обслуживание ЭВМ, как, например, ремонт телевизоров. А пока в высоких инстанциях решается вопрос, кто должен обслуживать компьютеры в зубе, простаивают и без того дефицитные персональные ЭВМ.

Несомненно, приобщая будущих специалистов к работе на ЭВМ, надо расширять сферу общения студентов с компьютером. Необходимо, чтобы они работали с ним в лабораториях на кафедре при курсовом и дипломном проектировании. Тогда у студентов появится вкус к вычислительной технике и они станут искать возможности дополнительно поработать с ЭВМ при изучении нового учебного материала. В рамках студенческого научного общества мы прививаем специалистам навыки автоматизации исследовательской работы с помощью ЭВМ. Например, ученые нашего зуба совместно с другими научными институтами разработали оригинальную автоматизированную систему контроля качества и состава материалов «Атлант». В ее основе — микро-ЭВМ, которая проводит анализ тончайших срезов металлов, пластмасс и других материалов. Телекамера передает их изображение на компьютер. Он анализирует структуру исследуемого среза и сообщает экспериментатору необходимые данные и параметры. Результаты измерений выводятся на экран дисплея и печатаются принтером. Каждое изображение среза анализируется компьютером от одной до десяти секунд.

Система «Атлант» автоматизирует малопродуктивный труд по анализу структур материалов, сокращает время исследований, улучшает их качество.

Используя такую систему в зубовской лаборатории, студенты убедятся в преимуществе вычислительной техники, станут пропагандировать ее внедрение в институтах и на заводах, куда они придут после окончания зуба. Но тут возникает еще одна неувязка. Иногда молодые специалисты не могут найти применение тем знаниям, которые они получили в зубе. Например, некоторые наши выпускники жалуются, что на предприятиях, куда они направляются на работу, вычислительная техника используется вполсилы либо вообще отсутствует. В этом случае мы сталкиваемся с острой проблемой эффективного использования на производстве знаний молодых специалистов. Если зуб при их подготовке старается широко внедрять вычислительную технику, а на предприятиях, куда

распределяются выпускники, о ней знают понаслышке, грош цена нашим стараниям подготовить технически грамотного, современного специалиста, требовать от него работы с полной отдачей знания и сил.

Сегодня хозяйственникам пора уже понять, что ЭВМ не роскошь, а современный инструмент в руках специалистов. И чтобы он действительно повышал производительность инженерного труда, необходимо объединить усилия институтов и предприятий в совершенствовании компьютерного обучения.

М. ЖЕБРАК

Использование ЕС ЭВМ в обучении

Использование в учебном процессе существующего парка вычислительной техники, которой оснащены предприятия и учреждения нашей страны, — один из перспективных подходов к проблеме компьютеризации образования. Огромные вычислительные ресурсы, возможность подключения к ЭВМ десятков внешних устройств, находящихся на значительном удалении от ВЦ, создание сетей ЭВМ — все это поможет становлению нового учебного предмета — основы информатики и вычислительной техники, а также обеспечит высокую эффективность учебного процесса в целом.

В 1985 г. в Хабаровском индустриальном техникуме создан КВТ, оборудованный дистанционным комплексом ЕС-7920—11. Посредством стандартных устройств сопряжения комплекс связан с ЭВМ ЕС-1052, установленной в ВЦ Дальневосточного научного центра АН СССР. Опыт двух лет эксплуатации кабинета показывает, что на общем фоне решаемых задач учащиеся не создают помех профессиональным пользователям.

Возможности комплекса ЕС-7920—11 соответствуют возможностям центральной ЭВМ. Таким образом, учащиеся получают доступ к существующим пакетам прикладных программ. В дальнейшем намечается подключение к Академсети АН СССР.

Программное обеспечение КВТ осуществляется силами базового ВЦ. В настоящее время функционирует система диалогового доступа ROSCOE, которая предоставляет преподавателю и учащимся довольно широкие возможности.

На базе ROSCOE могут быть разработаны обучающе-информационные системы; в пер-

Ясно, что сегодняшние выпускники вузов должны уметь работать с ЭВМ. А как обеспечить институты страны необходимой вычислительной техникой, пока не ясно. Кто должен отвечать за техническое обслуживание и математическое обеспечение вузовских компьютеров? Единого мнения у министерств и ведомств по этому вопросу нет. Пора преодолеть межведомственные барьеры и создать такие организации, которые бы полностью отвечали за компьютеризацию обучения в вузе.

77
спективе возможен доступ из ROSCOE уже готовым обучающим программам, в среде ROSCOE имеется возможность программирования на наиболее распространенных в настоящее время языках.

Возможности, предоставляемые системой, составляют три группы. Первая группа включает подсистемы, обеспечивающие программирование на языках Бейсик, Фортран, ПЛ, Паскаль. При этом полностью исключена необходимость изучения языка управления заданиями и особенностей операционной системы, что до сих пор значительно затрудняло использование ЕС ЭВМ в учебных целях.

Вторая группа обеспечивает собственно процесс обучения. В рамках этой группы разработана автоматизированная обучающая система, представляющая собой некоторый аналог существующей АОС ВУЗ. Система позволяет изучать теоретический материал по основам информатики и вычислительной техники и некоторым другим предметам, а также осуществляет контроль знаний. Весьма перспективным представляется внедрение АОС ВУЗ. В среде АОС ВУЗ созданы сотни обучающих и обучающе-контролирующих программ, что позволяет раздвинуть рамки применения ЭВМ на абсолютное большинство изучаемых предметов.

Следующая группа обеспечивает возможности управления. Информационно-поисковая система и Записная книжка директора — только первый шаг в этом направлении. В дальнейшем предусмотрено внедрение автоматизированной системы управления. Разработку первой очереди АСУ учебным процессом выполнил ВЦ Белгородского технологического института строительных материалов. Система обеспечивает накаплива-

ние информации о текущей успеваемости и посещаемости и оперативное предоставление указанной информации в любом разрезе: по группам, по курсам, конкретным учащимся. Подсистемы «Абитуриент», «Контингент» и другие разработаны также Белгородским институтом и могут быть легко адаптированы для техникума. Подсистемы, входящие в рассматриваемую группу, позволяют интенсифицировать учебный процесс и высвободить время преподавателей и административных работников.

Методика преподавания курса ОИВТ в Хабаровском индустриальном техникуме обусловлена приведенной выше структурой программного обеспечения. Теоретические занятия по ключевым темам проводятся в КВТ с использованием АОС «Техникум». Для проведения практических занятий разработан лабораторный практикум (34 ч). Действующей программой по ОИВТ не раскрывается содержание практических работ. Разработанный лабораторный практикум ориентируется на программу конкурсного учебника информатики (3) и включает следующие работы:

1. Оборудование кабинета вычислительной техники. Правила техники безопасности при работе на ЭВМ.

2. Знакомство с клавиатурой ЭВМ.

3. Начальные навыки работы на ЭВМ.

4. Составление и исполнение алгоритма вычисления математической функции.

5. Составление и исполнение алгоритма работы с таблицами.

6. Двоичное кодирование информации.

7. Работа с базовым программным обеспечением.

8. Составление и отладка линейной программы средствами языка Бейсик.

9. Составление и отладка разветвляющейся программы средствами языка Бейсик.

10. Составление и отладка циклической программы средствами языка Бейсик.

11. Сопоставление и отладка программы, содержащей подпрограммы.

12. Разработка программ решения задач из курса математики и физики.

13. Разработка программ экономических расчетов.

14. Работа с текстовым редактором.

15. Работа с базой данных.

16. Работа с автоматическим решателем задач.

17. Решение учебно-производственных задач на ЭВМ.

Учебное программное обеспечение, входящее в состав АОС «Техникум», разрабатывается учащимися в ходе курсового и дипломного проектирования. Привлекаемые к этой работе студенты техникума проходят практику в Региональном центре по обслуживанию ЭВМ или в ВЦ ДВНЦ АН СССР. В настоящее время созданы программы по электротехнике, математике, экономике, автоматизации производственных процессов.

Сейчас отдельные элементы программного обеспечения дорабатываются и апробируются в условиях учебного процесса. Готовятся методические указания по выполнению лабораторных работ по ОИВТ, а также пособие по дипломному проектированию с применением ЭВМ.

Решение проблемы компьютерной грамотности неразрывно связано с производством. Накопленный в техникуме опыт позволяет ставить вопрос о технической помощи предприятиям отрасли в автоматизации технологических процессов, разработке новых материалов для строительства. В студенческом конструкторском бюро создаются проекты, осуществлять которые будут сами учащиеся во время производственной практики. Сейчас решается вопрос о разработке программ по заказам предприятий на условиях хоздоговора.

Литература

1. О мерах по обеспечению компьютерной грамотности учащихся средних учебных заведений и широкого внедрения электронно-вычислительной техники в учебный процесс: Сборник документов и материалов. М., 1985.

2. Изучение основ информатики и вычислительной техники: Методическое пособие для учителей и преподавателей средних учебных заведений / Под ред. А. П. Ершова, В. М. Монахова. М., 1985. Ч. I, II.

3. Программа курса «Основы информатики и вычислительной техники» // Математика в школе. 1986. № 3.

Программируемые микрокалькуляторы в педагогическом институте

Существуют различные, порой противоположные, точки зрения относительно возможности и целесообразности применения программируемых микрокалькуляторов (ПМК) в процессе подготовки учителей математики, физики и информатики. Что касается наличия современных вычислительных средств в педагогических вузах страны, то некоторые институты имеют персональные компьютеры, в других же нет и программируемых микрокалькуляторов.

Применение ПМК в учебном процессе мы считаем безусловно целесообразным. Эта позиция — не «от бедности»: наш педагогический институт обеспечен персональными компьютерами. Она вытекает из трезвой оценки ближайшей перспективы развития вычислительной базы в школах республики, в которых предстоит работать нашим выпускникам. Нам представляется, что центральные органы, распределяющие вычислительные средства в системе Минпроса, неминуемо придут к выводу о необходимости скорейшего обеспечения всех школ достаточным числом ПМК. Соответственно должна состояться и частичная переориентация школьного курса информатики. В нем необходимо предусмотреть обучение программированию на микрокалькуляторах. Несомненно, что ПМК, используемый как подспорье в изучении специальных дисциплин, дает возможность сократить время рутинной работы и сосредоточиться на анализе различных вариантов решения, а также закрепить знание основных численных методов.

Для нас существует подбор задач именно для педагогического института. Приоритет следует отдавать задачам, в процессе решения которых при минимуме необходимых численных методов достигается максимум дидактической глубины и ясности, возможны различные решения, а также чередование счета и качественного анализа.

Такова общая концепция. Теперь о некоторых практических результатах.

У нас установились следующие основные формы организации обучения математике, физике и астрономии с использованием ПМК: аудиторная работа, кружки, индивидуальная работа.

Аудиторная работа. Случай применения ПМК на занятиях по теоретической физи-

ке — использование стека и регистров для вычисления различных комбинаций физических постоянных. Необходимый при этом анализ размерности и физического смысла вычисляемых величин — важный элемент подготовки школьного учителя. Таким образом студенты могут познакомиться с характерными атомными масштабами, постоянной тонкой структуры и т. д.

Пример более сложной задачи из квантовой механики: рассчитать подробно коэффициент прохождения через потенциальный барьер. Для самостоятельной работы предлагается составить разветвляющуюся программу (подбарьерный и надбарьерный случаи) при различных числовых данных.

Особенно полезны задачи, сочетающие качественный анализ, графическое решение и численный метод. Примером может служить расчет энергии связи дейтона в известном приближении прямоугольной потенциальной ямы. В аудиторной составляется уравнение Шредингера и качественно анализируется спектр. Полученные из граничных условий уравнения для собственных значений решаются на ПМК с индивидуальными параметрами.

На занятиях по ядерной физике ПМК позволяет резко сократить время, затрачиваемое на вычисление энергии реакций, дефекта масс и т. п., требующее удержания многих знаков и использования справочных данных.

Курс общей астрономии включает несколько разделов. Каждый из них оперирует своим математическим аппаратом, что значительно усложняет усвоение материала. Обилие математических методов и громоздкость вычислений приводят к тому, что времени на объяснение собственно астрономических явлений остается мало. Применение на занятиях ПМК позволяет преодолеть некоторые из этих трудностей.

Решение задач астрометрии требует наличия астрономического календаря. Координаты Луны, планет и Солнца вычисляются методами небесной механики. Однако наличие стандартной программы вычисления эфемерид позволяет с достаточной для учебных целей точностью вычислить положение не только указанных тел, но и астероидов, комет.

Стандартная программа вычисления масс небесных тел по третьему обобщенному закону Кеплера позволяет за короткое время вычислять массы всех планет, имеющих спутники, массу Солнца, а также определять массы двойных звезд в астрофизике.

Легко программируются задачи по определению времени космических перелетов, начальных и конечных скоростей летательных аппаратов. Представляет интерес и стандартные программы по астрофизике звезд, позволяющие определять их эффективную температуру и, в первом приближении, внутреннее строение.

Перечисленные задачи и соответствующие стандартные программы могут быть использованы и в кружковой работе. Заслуживает внимания идея публикации стандартных программ в методических пособиях, в периодической методической литературе, а также отдельными брошюрами.

80 Кружки. Старшекурсники, объединенные в кружок теоретической физики, по завершении программы кружка выбирают темы для курсовых и дипломных работ, связанные с использованием ПМК.

На занятиях кружка задачи решаются коллективно, под руководством преподавателя или наиболее сильных студентов. В качестве домашних заданий просчитываются варианты задач, дописываются и отлаживаются программы и т. п.

Особое внимание хотелось бы обратить на циклы задач, связанных с одним физическим объектом. Они позволяют всесторонне изучить определенный раздел физики и одновременно освоить несколько численных методов. Показателен в этом отношении цикл задач по теме «Равновесное излучение». Сюда можно включить определение температуры по экспериментальным данным (статистическая обработка), вычисление постоянной Вина и аналогичных констант (задачи на экстремум), вычисление постоянной Стефана (численным интегрированием или суммированием ряда).

Программа математического кружка для первокурсников рассчитана на два месяца первого учебного семестра. Студенты учатся программировать на калькуляторе, решая несложные алгебраические и геометрические задачи школьного типа. Подача материала построена по принципу: определению той или иной новой команды всегда предшествует формулировка специально составленной элементарной задачи. Затем анализируются пути решения этой задачи. Возникает проблемная ситуация, исход которой — осознание необходимости новой команды, которая тут же вводится и разбирается. Так, например, определению

команды условного перехода предпосылается задача о вычислении уровня воды в водокачке при заданном объеме воды и заданных параметрах водокачки, а определению перехода на подпрограмму — задача о вычислении площади шестиугольника по заданным диагоналям, исходящим из одной вершины, и сторонам.

На последних занятиях разбираются три стандартные программы: построение графика функции, решение уравнения методом деления пополам и нахождение точки экстремума функции. Эти программы отличны от тех, что приводятся в пособиях по ПМК. Они разработаны с учетом максимальной простоты приемов программирования, чтобы будущие учителя могли их непосредственно применять на уроках и факультативных занятиях в школе.

Индивидуальная работа. В начале второго семестра математический кружок для первокурсников вливается в общий семинар. Остальные участники семинара — кружковцы прошлых лет. Работа его строится по принципу индивидуальных заданий. На заседаниях обсуждаются решения задач участниками семинара, постановка новых задач.

Пока первокурсники не приобрели навыки самостоятельной работы, им приходится уделять особое внимание. Ко времени итоговой вузовской научной конференции (апрель) большинство студентов первого курса могут вполне уверенно, четко и аргументированно изложить перед аудиторией обоснование выбора задач, методы решения, проанализировать трудности.

Первокурсники фактически не владеют основами дифференциального интегрального исчисления, а ведь это традиционная область применения вычислительных средств. Мы нашли выход: студенты решают на ПМК задачи школьного типа, не допускающие аналитического решения. (Возможно, что такое решение существует, но выходит за рамки школьной программы.)

Выделим несколько групп таких задач. Во всех случаях речь идет, разумеется, о приближенном решении с заданной точностью.

1. Задачи, сводящиеся к решению уравнений высокой степени или трансцендентных уравнений. Таковы некоторые задачи на соразмерность длин окружностей и периметров многоугольников, площадей круглых и многоугольных фигур, задачи на определение параметров сечений пространственных тел и др.

2. Задачи, сводящиеся к нахождению точки минимума или максимума функции (предполагается, что приравнивание производной к

нулю приводит к уравнению высшей степени или трансцендентному уравнению). Такими являются некоторые задачи на оптимизацию объемов, площадей, сумм расстояний углов и др.

3. Физико-геометрические задачи. Рассматриваются разнообразные «качели», представляющие собой механические системы с одной степенью свободы, состоящие из невесомых стержней с шарнирами и пружин и включающие по одной материальной точке. Требуется определить параметр системы (обычно какой-нибудь угол) в состоянии равновесия. Учитывается действие силы тяжести.

Решение перечисленных задач на ПМК основывается исключительно на применении стандартных программ решения уравнения и нахождения точки экстремума функции. Третья упомянутая выше стандартная программа построения графика используется в случае необходимости дополнительно исследовать функцию.

Задачи более высокого уровня связаны с определением метода колебаний, описанных в п. 3, и требуют применения стандартной программы интегрирования. Это относится также к задаче об определении уровня, на который поднялась вода в озере после прохождения дождевой тучи (заданы все необходимые параметры озера, а также тучи, ее движения), и др.

Теперь об индивидуальной работе со студентами, связавшими с ПМК темы своих

курсовых и дипломных работ. Мы стремимся, чтобы каждая работа была пусть небольшим, но самостоятельным исследованием. По-видимому, назрела необходимость в масштабе страны разработать единую тематику таких работ для однотипных специальностей. Это облегчило бы организацию самостоятельной работы и создало бы организационную основу для конкурсов и конференций разного уровня.

В качестве тем для дипломных работ кафедрой теоретической физики предлагались, например, следующие. «Вариационные расчеты простейших атомов и молекул», «Вычисление термодинамических функций по энергетическому спектру».

Дипломные работы по астрономии связаны в основном с задачами небесной механики и астрометрии. После первых публикаций о предстоящей встрече с кометой Галлея возникла задача определения орбиты, видимости кометы в условиях Тирасполя, возможности проведения ее наблюдений при имеющихся ограниченных оптических средствах. Другая тема связана с осуществлением программы «Фобос». Была поставлена задача об условиях перелета к Марсу и задача составления карты звездного неба для наблюдателя на Марсе.

Относительно низкая стоимость, доступность, перспективы развития ПМК позволят им успешно конкурировать с микро-ЭВМ (учитывая возможность подсоединения ПМК к телевизору и бытовому магнитофону).

81

ИНФОРМАЦИЯ

Конференция в Минске

Специалисты, собравшиеся с 15 по 17 сентября 1987 г. в Минске на первую Всесоюзную конференцию по проблемам создания супер-ЭВМ, посвятили это мероприятие памяти выдающегося советского ученого С. А. Лебедева, 85-летие которого отмечалось в этом году.

С докладами выступили академики В. А. Мельников, К. А. Валиев, президент АН БССР В. П. Платонов.

В конференции приняли участие представители основных научных центров страны.

Была отмечена необходимость тесного взаимодействия и объединения усилий сформировавшихся научных школ страны и выделены некоторые важные направления:

- разработка и создание новых технологий и материалов,
- создание ЭВМ новых архитектур,
- исследование теоретических основ организации параллельных вычислений,
- создание методов конструирования надежных систем.

Участники заседаний выразили пожелание сделать такие конференции традиционными.



СТРОКИ ИЗ ПИСЕМ

Журнал «Информатика и образование» сразу завоевал популярность среди всех категорий читателей, причастных к обучению информатики учащейся молодежи. Первые номера журнала передавались из рук в руки, как обычно передается журнал с нашумевшим литературным произведением. Интерес к новому периодическому изданию велик как среди работников вузов, обучающих учителей, так и работников предприятий, оборудующих КВТ в школе или помогающих проводить практические занятия на своих вычислительных центрах, и, конечно, самое главное, — среди учителей информатики. Это не просто общие фразы, а наши конкретные наблюдения на протяжении всего времени, прошедшего с выхода первого номера, так как нам постоянно приходится иметь дело со всеми категориями специалистов. Интерес понятен — создан не просто новый предметный журнал. Это печатный орган, знаменующий вступление школы в новый этап своего развития — компьютеризацию, или, более широко, информатизацию, которая началась пока введением курса «Основы информатики и вычислительной техники».

Материалы первых номеров, посвященные вопросам цели, стратегии и тактики компьютеризации школы, способствуют выработке правильной ориентации среди всевозможных мнений, предложений, инициатив профессионалов, взявшихся за это совершенно новое для школы дело.

На страницах журнала отражаются сегодняшний опыт и проблемы школьной информатики, поэтому он способствует повышению квалификации и учителя, приступившего два года назад к освоению информатики, и профессионала, передающего свои знания школьникам.

Методика преподавания ОИВТ только начинает создаваться, и поэтому журнал, публикующий методические материалы, приобретает чрезвычайную важность в становлении нового предмета.

Членами научно-методической комиссии области, приступающей к формированию программного обеспечения школьной техники области, пристально изучаются материалы по математическому обеспечению, требования к педагогическим программным продуктам и методике их использования на уроке.

Журнальные публикации используются при подготовке дипломных работ выпускниками механико-математического факультета Саратовского государственного университета — будущими педагогами.

«Информатика и образование» исследует все проблемы, возникающие на первом этапе информатизации школы, пишет о видимых сегодня путях их решения, обсуждает как результаты конкретных дел, так и перспективы дальнейшего развития, и поэтому читательская аудитория журнала будет постоянно увеличиваться.

Поэтому хотелось бы видеть журнал, полигра-

фическое оформление которого соответствовало бы его назначению и обеспечивало бы возможность многократного обращения к каждому номеру в течение продолжительного времени.

Нужны, лучшая бумага, прочные обложки и иллюстрации высокого качества.

Н. АНДРЕЕВА

Канд. физ.-мат. наук, ст. преподаватель

мехмата СГУ;

Н. КАЗАКОВА

Методист ОИУУ, Саратов

Уважаемые товарищи!

Пишет вам начинающий учитель курса ОИВТ. Я окончил пединститут в 1964 году, когда об информатике и речи не было. И вот теперь волею судьбы должен преподавать ее в школе. Сам я плохо ориентируюсь в этой области. Съездил на краткосрочные курсы в областной ИУУ, где нам объяснили основы Бейсика и Фортрана и больше ничего. Таких пожилых учителей, как я, на курсах было большинство. Все, как и я, не имеют никакого понятия об этом предмете. Во многих школах нет даже микрокалькуляторов, не говоря уже о ПЭВМ.

Я пишу все это к тому, что ваш журнал таким, как я, ничем не помогает. Очень мало конкретных методических материалов, показывающих, как можно изложить ту или иную тему. Почти ничего не печатается о работе с программируемыми микрокалькуляторами. Большинство материалов рассчитано на подготовленного читателя, хорошо разбирающегося в алгоритмических языках и в вычислительной технике. А таких учителей мало.

Поэтому — просьба: повернуться лицом к школе, печатать больше методического материала, который помог бы таким учителям, как я.

В. БЛИНОВ.

с. Бобино Слободского р-на
Кировской обл.

Следует постоянно привлекать к активной работе со школьниками вузовский комсомол.

Я предлагаю организовать для студентов математических специальностей пединститутов и университетов производственную практику по преподаванию информатики. И проводить ее в течение одного или нескольких семестров без отрыва от учебы. Разумеется, работу студентов нужно оплачивать.

Большую пользу, на мой взгляд, могли бы принести комсомольско-молодежные хозрасчетные УПК при институтах, выполняющие работу в основном производственного характера. Так, они могли бы выполнять расчеты для небольших предприятий (магазины, предприятия службы быта и общественного питания и многие другие), не имеющих своих информационно-вычислительных центров.

А. АШКИНУЗЕ,

Москва

Журнал в журнале
для школьников, студентов,
учащихся СПТУ и техникумов

Издается при участии
ЦК ВЛКСМ

МОЛОДЕЖНАЯ ИНИЦИАТИВА

1

В НОМЕРЕ:

-
- 1000 ПРОФЕССИЙ ЭВМ
-
- СЦЕНАРИЙ ПРОГРАММЫ, НАПИСАННЫЙ УЧЕНИКОМ VIII КЛАССА
-
- КРУЖОК ЭЛЕКТРОННОГО КОНСТРУИРОВАНИЯ: 2 ЗАНЯТИЯ
-
- ЭТО ВЫ МОЖЕТЕ КУПИТЬ В МОСКВЕ НА ЛЕНИНСКОМ ПРОСПЕКТЕ
-
- КТО ПОДАРИТ КОМПЬЮТЕР СЕЛЬСКОМУ ШКОЛЬНИКУ!
-

1000 профессий ЭВМ

О конференции школьников «1 000 профессий ЭВМ» рассказали И. Я. Ланина и А. Л. Лернер. Надеемся, что эта публикация поможет нашим читателям провести конференцию в своей школе, техникуме, училище.

ЭВМ — универсальный преобразователь информации, необходимый людям разных профессий: и переводчику, и математику, и поэту, и рабочему. Цель школьной конференции «1 000 профессий ЭВМ» — на конкретных примерах показать, какую роль в современной жизни играют кибернетика и вычислительная техника. Такие конференции можно с успехом проводить как в IX-X классах, так и в VI-VIII, изменяя в зависимости от возраста учащихся содержание докладов.

В школе № 248 Ленинграда подготовка к конференции «1 000 профессий ЭВМ» началась с объявления сбора научной информации, который проводили в четырех направлениях: подбор литературы, ее изучение; экскурсии — знакомство с ЭВМ определенной профессии

(учащиеся школы посетили цех завода, где работают станки с ЧПУ, диспетчерскую метрополитена, гидрометеорологический центр, лабораторию математической лингвистики, больницу, агрофизический институт); составление рефератов для выступления на конференции; участие в конкурсе на лучший проект новой ЭВМ.

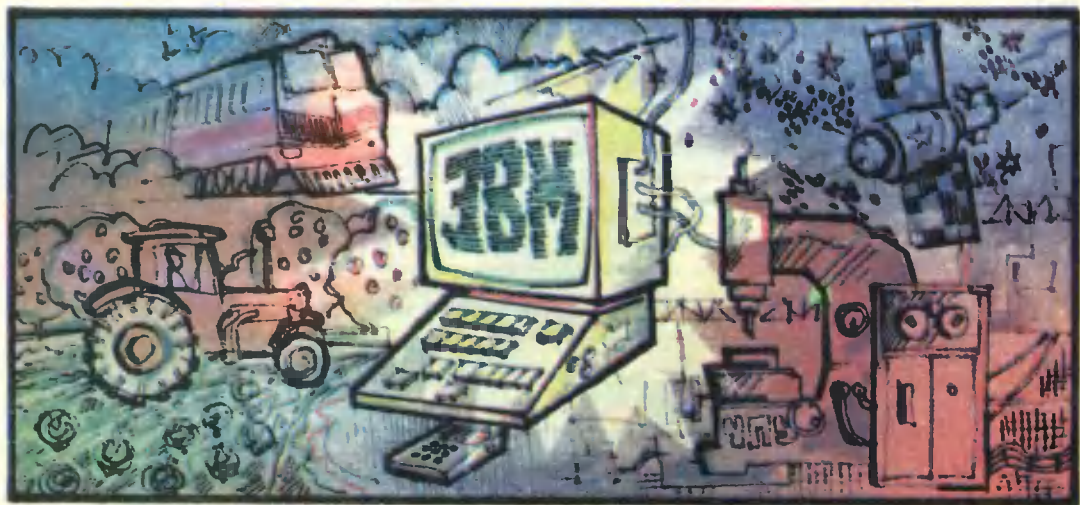
Равноправным участником конференции была «сконструированная» учащимися «машина пятого поколения», «снабженная» синтезатором речи, дисплеем, вводным устройством, постоянной памятью, блоком управления и выводным устройством. Изготовить макет машины нетрудно. Он состоит из двух основных частей — лицевой панели и каркаса. Лицевая панель имитирует пульт управления машины. Здесь же вырезают круглые отверстия и заклеивают их цветным целлофаном. На обратной стороне листа, напротив отверстий, укрепляются лампочки «мигающей» елочной гирлянды. Каркас, удерживающий лицевую панель и все остальные устройства, можно собрать из школьных

штативов. Экраном дисплея служит матовое стекло (60×60 см) или лист плотной бумаги, укрепленный в лапах штативов. Изображение на стекле создается с помощью кодоскопа, устанавливаемого за макетом машины и невидимого зрителям. Каждая из кодограмм, подготовленных заранее для диалога докладчика с машиной, содержит только одну часть информации. В ходе доклада пленки кодограмм постепенно накладываются друг на друга.

Важными частями макета являются электрический мегафон «Рубин» и электронный счетчик-секундомер, установленный в режиме счета времени. Блок памяти представляет собой две бобины от киноаппарата или магнитофона со вставленной между ними перфолентой. Во время «работы» машины бобины вращаются.

Конференцию открыли шуточные сценки — истории ЭВМ разных поколений.

II. ЭВМ сладко спит, а уставший программист пишет программу (составление программ для машин перво-



го поколения занимало 98 % рабочего времени).

II. Программист стоит перед книжными полками, заставленными томами с названиями «Алгол», «Фортран», «Кобол» и другими, а ЭВМ хохочет в стороне (с появлением машин второго поколения буквально за 5—6 лет возникло около 500 алгоритмических языков. Сейчас их около 1 000).

III. Несколько программистов в поте лица трудятся, а ЭВМ разбирает и отдает им перфоленты (машины третьего поколения могут обрабатывать до 15 программ одновременно).

IV. Двое программистов в микроскоп разглядывают ЭВМ, а третий подает на чашечке Петри крошечные программы (благодаря применению полупроводниковых элементов и интегральных схем размеры ЭВМ значительно уменьшились).

Далее школьники, выступающие как представители различных предприятий и учреждений, познакомили участников конференции со своими докладами. Предлагаем один из них вашему вниманию. Функции ведущего «исполняла» ЭВМ.

ЭВМ. Внимание, с вами говорит ЭВМ пятого поколения, оснащенная синтезатором речи. Приветствую всех участников конференции «1 000 профессий ЭВМ». Рабочий язык симпозиума — русский. Одновременно даю перевод на французский, английский и немецкий языки. Прошу наших иностранных гостей надеть наушники. (Перевод осуществляется на тот иностранный язык, который изучается в школе.)

Для получения хороших урожаев большое значение имеет прогноз погоды. Послушаем доклад о машине-метеорологе. Слово Игорю Майкову, работнику гидрометеорологического центра.

Докладчик. По всему зем-

ному шару метеорологи несколько раз в сутки фиксируют температуру воздуха, давление, направление и силу ветра... Эти данные подаются в метеоцентры, где синоптики сводят их воедино, собирая из тысячи данных, как из кусочков мозаики, цельную картину погоды. Линиями изобар и изотерм расчерчиваются синоптические карты, и вот уже воочию видно, где сегодня образовался циклон или антициклон, куда и с какой скоростью он движется, неся с собой потепление или похолодание, хорошую или плохую погоду. (Докладчик демонстрирует карту.)

Казалось бы, имея такие данные, прогнозировать погоду на завтра и послезавтра довольно просто. Но простота эта обманчива. Антициклон может свернуть в сторону и изменить свою скорость, и то и попросту прекратить свое существование... Чтобы избежать ошибок, специалисты вводят в ЭВМ все имеющиеся данные, допустим, о том же антициклоне. И машина, руководствуясь этими данными и уже известными законами погодообразования, заложеными в программу, рассчитывает возможные варианты.

Тысячи метеостанций в нашей стране, по территории которой проходит 13 часовых поясов, собирают сведения о погоде (показывает на карте СССР или мира).

Из космоса информацию передают на землю более 600 спутников. Оперативно обработать это огромное количество данных невозможно не только вручную, но и с помощью самой большой быстройдействующей ЭВМ. Нужна целая сеть вычислительных машин, занимающихся прогнозом погоды. При этом большая часть информации должна обрабатываться на месте, в ме-

теоцентр будут передаваться лишь обработанные результаты, как их называют математики, пакеты информации.

Прогноз погоды должен быть очень точным и по возможности долгосрочным. Это важно везде, особенно в сельском хозяйстве, в авиации, где любая неточность может повлечь за собой катастрофу. Поэтому ученые увеличивают число данных, на основе которых составляется прогноз. Разработана специальная программа «Разрез». В заранее выбранных районах Мирового океана исследователи проводят замеры, начиная с космических высот и кончая дном океана.

Такие измерения очень важны, поскольку большая часть Земли покрыта водой, а циклоны и антициклоны, бури и тайфуны, которые «делают» погоду на планете, зарождаются над океаном.

Кроме того, на формирование погоды огромное влияние оказывают синоптические вихри — гигантские водовороты в океане. Их диаметр — до нескольких сотен километров, поэтому они могут длиться месяцами.

ЭВМ, получая данные с поверхности и из глубин океана, установит их взаимосвязь с процессами в атмосфере и сможет точно предсказать погоду на месяц вперед.

Например, я прошу машину выдать данные о погоде на 25 июля 1988 года в точке 43° 30' восточной широты. (ЭВМ выдает данные на дисплей.)

Итак, ЭВМ, становясь метеорологом, должна помочь побыстрее собрать нужное количество информации о погоде, обработать ее и передать результаты обработки всем заинтересованным пользователям. Найдены принципиальные возможно-

сти решения этой задачи. Информация может быть оперативно собрана при помощи метеостанций, кораблей погоды, искусственных спутников Земли... Ее обработкой должны заниматься не люди, а вычислительные машины — так будет быстрее. Необходимо пересылать пакеты информации «снизу вверх» и «сверху вниз», используя самые быстрые каналы связи — радио, телеграф, телефон...

Данные пакетов позволяют ЭВМ самой чертить точные синоптические карты. (Машина выдает распечатки.)

На конференции были также заслушаны доклады об ЭВМ-диспетчере, ЭВМ-переводчике, ЭВМ, имеющей профессию врача, об использовании ЭВМ в сельском хозяйстве и др.

На вопросы, обращенные к ЭВМ из зала, отвечали докладчики или учитель, находящийся за панелью макета. При первом вопросе на который «машина» ответить не сможет, обнаруживаются неполадки: шум, быстрая смена цифр на табло секундомера.

Вторая часть конференции была посвящена подведению итогов конкурса школьников на лучший проект ЭВМ.

Учащиеся школы № 248 представили для защиты 6 интересных проектов: электронная машина времени; ЭВМ — справедливый экзаменатор; ЭВМ, управляющая системой работ для уборки в квартире; ЭВМ, управляющая продажей товаров в универсаме; ЭВМ, управляющая движением судов; производственный робот-универсал.

Конференцию можно завершить играми с микрокалькулятором по заранее составленным программам.

Рекомендуемая литература

1. Бусленко В. Н. Наш кол-лега — робот. М., 1984.

2. Громко Н. И. Введение в страну ЭВМ. Минск, 1984.

3. Ломов Б. Ф. Человек и автоматы. М., 1984.

4. Максимович Г. В. Беседы с академиком Глушковым. М., 1978.

5. Моисеев Н. Н. Слово о научно-технической революции. М., 1985.

6. Овчинкин В. В мире компьютеров. М., 1985.

7. Пекелис В. Д. Маленькая энциклопедия о большой кибернетике. М., 1973.

8. Рассказ об ЭВМ. М., 1984.

9. Романовский Т. Б. Микрокалькулятор в рассказах и играх. Рига, 1985.

10. Салтыков А. И., Семашко Г. Л. Программирование для всех. М. 1986.

11. Трохименко Я. К., Любич Ф. Д. Микрокалькулятор, Ваш ход! М., 1985.

12. Фомин Б. В. Автоматы служат человеку. М., 1980.

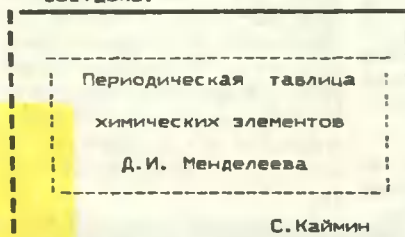
13. Щелкин А. Г. В наступающем «электронном веке». Л., 1982.

Сценарий программы «Химия»

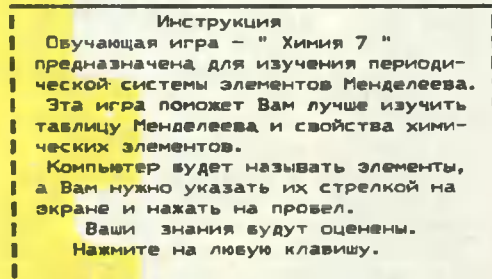
Составил ученик VIII класса С. Каймин

Сценарий программы "Химия 7"

Заставка.



Инструкция.



↓

Таблица.

Таблица Д.И. Менделеева 1869г.

H								He		
Li	Be	B	C	N	O	F	Ne			
Na	Mg	Al	Si	P	S	Cl	Ar			
Fr	Ra	Ac	Ku	Ns						

Укажите - «эл-т»

*

Повторение
10 раз

При нажатии на провёл.

При правильном ответе При неправильном ответе



«таблица-
Менделеева»

Это - «друг.-эл-т»
(По коор-там стрелки)

Модель Бора 1912г.

«обозн» атом:
«эл-т»

Свойства:

«изображе- номер: «н»
ние атома» слоев: «с»
вес: «в»
Кол-во
электронов
на слое;
внешнем: 2
внутр.:

При 10 кратном повторе
таблицы (в таблицу)

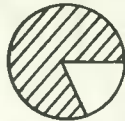


Твои
результаты



Правильно
90%

Общее число
правильных



ответов

оценка

11121314151

Нажмите на провёл

$K \equiv K+1$
если $K=77$ то ИНАЧЕ



Поздравляю вас с успешным
прохождением таблицы.



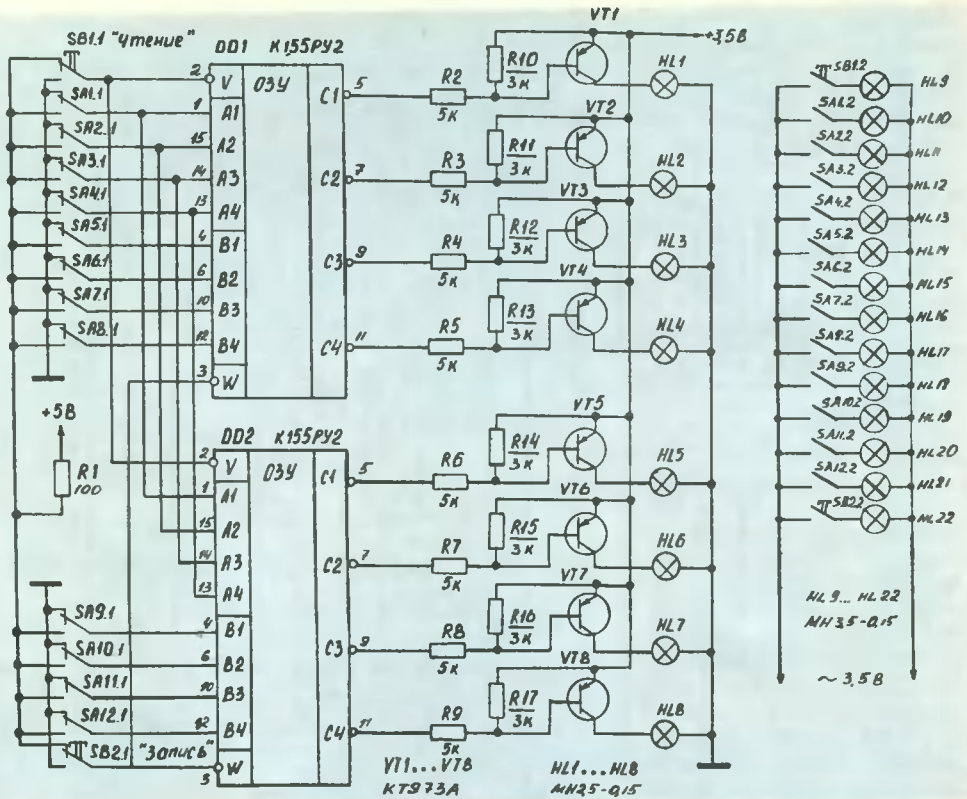
Пособия для уроков информатики

При изучении курса «Основы информатики и вычислительной техники» необходимо объяснять учащимся принцип работы основных узлов и блоков электронной вычислительной техники, устройств автоматики. Для этого можно использовать самодельные учебно-наглядные пособия, которые школьники могут выполнить на занятиях технического кружка. Одним из таких пособий может быть макет программного запоминающего устройства (ПЗУ). На схеме ПЗУ (рис. 1) набор горизонтальных (адресных) и вертикальных (разрядных) проводов. С его помощью можно записать и прочесть 4 слова по 8 разрядов. Диоды расположены во всех точ-

ках пересечения вертикальных и горизонтальных проводов. Последовательно с каждым диодом включена плавкая вставка с относительно большим удельным сопротивлением. Для включения устройства необходимо к клемме «—» подсоединить источник питания, например ИП-2, широко применяемый в школах. Клеммы «+» соединяем с клеммой напротив записываемого слова. Тумблеры SA1...SA8 включаем в тех местах, где необходимо записать «0». Нажимаем кнопку «запись» и регулятором напряжения, имеющимся на лицевой панели источника питания, по амперметру устанавливаем предельный ток 200—300 мА. При протекании такого тока

через вставку выделяется тепло, вследствие чего она расплавляется. Соответствующий диод окажется отключенным. Значит, в эти ячейки мы записали «0». В диоды, оставшиеся подключенными, соответственно — «1». Верхние лампы HL1...HL15 контрольные. Они загорают в момент нажатия кнопки «запись» и выключаются в момент удаления плавкой вставки. Для чтения слова нажимаем кнопку «чтение», и информационные лампы HL2...HL16 показывают порядок расположения «1» и «0» в слове. Наглядно видно, что процесс программирования ПЗУ сводится к удалению плавких вставок в необходимых местах. Ясно, что ПЗУ данного типа допускают только однократную запись информации.





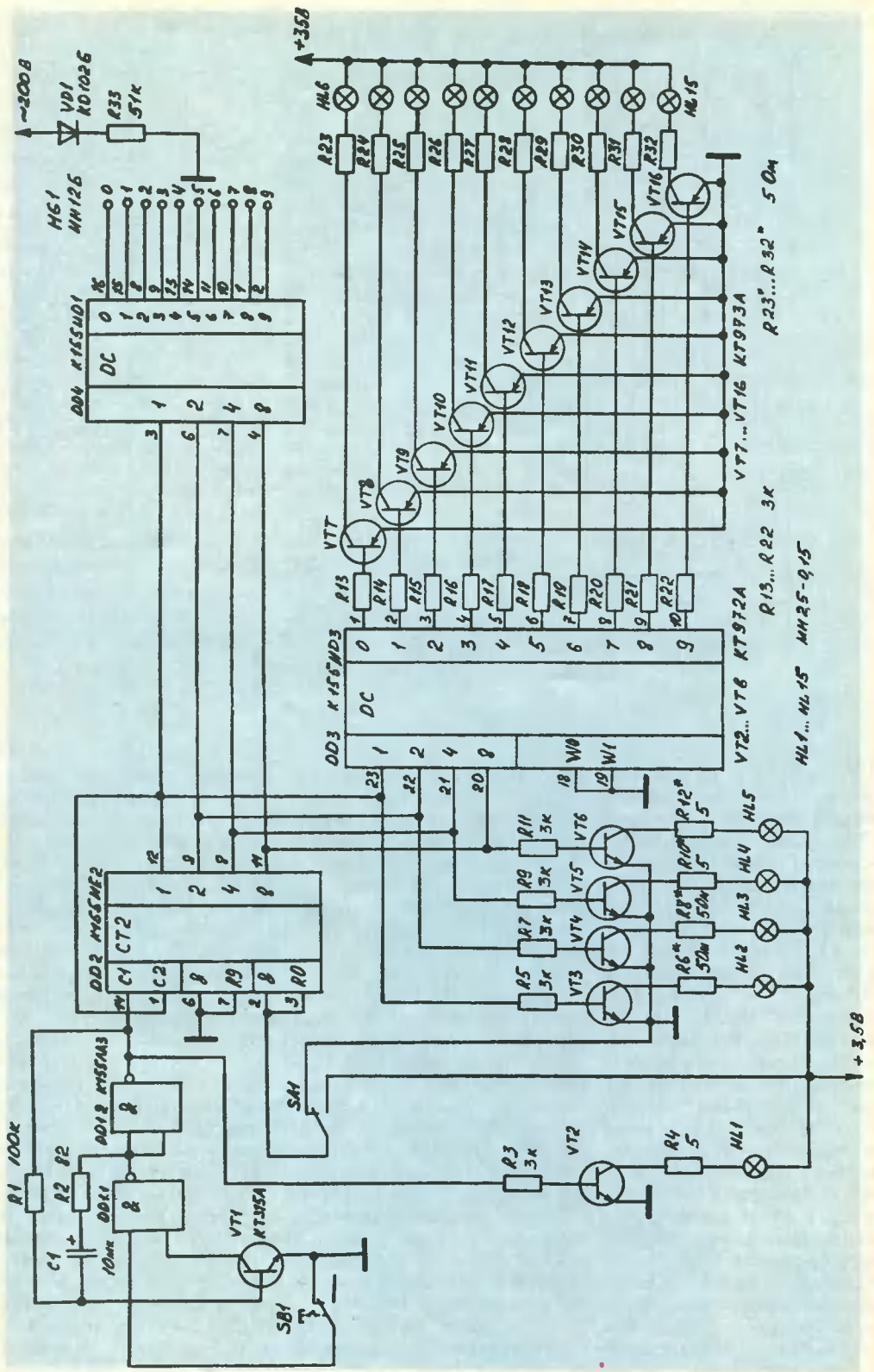
Кроме того, можно заполнить макет оперативного запоминающего устройства (ОЗУ). Оно предполагает многократное изменение записанной информации. Тумблерами SA1...SA4 (рис. 2) набирается адрес ячейки; SA5...SA12 информация, которую надо записать (лампы HL10...HL13 и HL14...HL21 индицируют адрес и входную информацию соответственно). Нажимаются кнопки «запись» и «чтение». При этом информация записалась, и тут же индицируется при помощи информационных ламп HL1...HL8, показывающих порядок расположения «1» и «0» в ячейках. При повторных демонстрациях информации в ячейках необходимо нажать кнопку «чтение». Хранение информации возможно лишь при подключенном электропита-

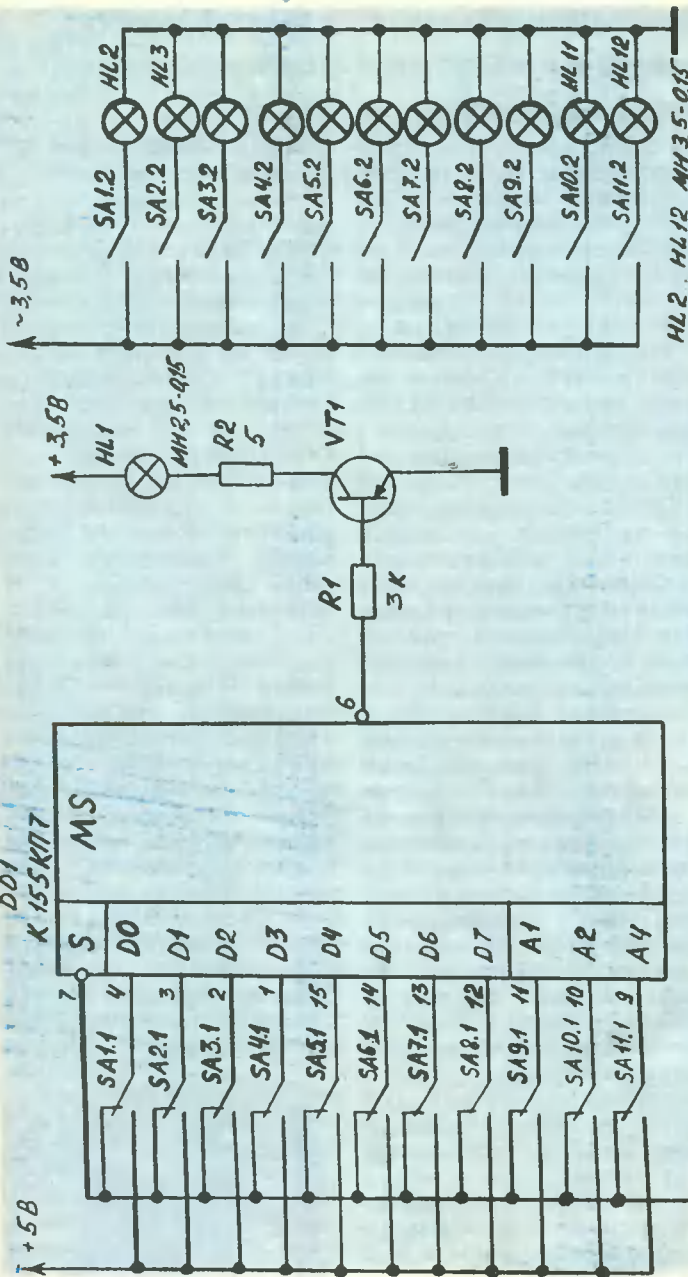
нии. Итак, ОЗУ — совокупность запоминающих ячеек, каждая из которых характеризуется своим адресом. Дешифратор и счетчик представлены на одном макете, принципиальная схема которого показана на рис. 3. Демонстрация начинается с установки счетчика в «0» тумблером SA1. При нажатии кнопки SB1 на счетный вход поступают импульсы частотой 1 Гц. Счетчик DD2 считывает импульсы, и информация на его выходах меняется в соответствии с количеством пришедших импульсов. Об этом можно судить по загоранию ламп. Таким образом счетчик подсчитывает импульсы, поступающие на его вход и фиксирует результат в виде двоичного кода на выходе.

Чаще всего дешифратор используется как преобразователь

двоичной информации в форму, удобную для включения цифровых индикаторов. В зависимости от заданного кода на одном из десяти выходов появится сигнал. Об этом можно судить по загоранию ламп HL6...HL15. Газоразрядный индикатор HG1 показывает число, которое имеется на выходе счетчика.

При передаче информации от нескольких устройств по одному каналу возникает необходимость в мультиплексоре. Это устройство, которое в зависимости от того, какой код набран тумблерами SA9...SA11 на адресных входах (рис. 4), разрешает прохождение сигнала на выход только с соответствующего из 8 информационных входов D0...D7. Входная информация индицируется лампами





HL2...HL9, а выходная HL1. Для электропитания используются такие номиналы напряжения: микросхемы — стабилизированный выпрямитель +5В, 250 мА; анод индикаторной лампы

счетчика-дешифратора — +200 В;
лампы ОЗУ — 3,5 В, 3 А;
лампы счетчика-дешифратора — 3,5 В, 1 А;
лампы мультимплексора — 3,5 В, 2 А.

Генератор счетчика-дешифратора собран на логических элементах DD1.1 и DD1.2 и транзисторе VT1.

Для повышения мощности сигналов на выходе микросхем используются транзисторные ключи. В схеме макета счетчика-дешифратора мы используем два дешифратора. Один — DD4 для управления газоразрядным индикатором, второй — DD3 для управления лампами.

Размеры корпуса каждого макета 800×500×100 мм. На передней панели макета ПЗУ есть съемное оргстекло 200×400 мм, где расположены диоды и предохранители. Их монтаж выполняется на диэлектрическом материале (текстолит, гетинакс). Для предохранителей используются зажимы (гнезда) из меди толщиной 0,3—0,5 мм, по типу тех, которые установлены в электроприборах. Можно вместо предохранителей использовать проволоку, диаметр которой определяется расчетом

$$d = I_n k + 0,05,$$

где d — диаметр проволоки в мм;

k — коэффициент (например, $k_{\text{меди}} = 0,034$);

I_n — ток плавления вставки.

Пример расчета и коэффициенты для различных материалов можно найти в справочнике «Элементы и детали радиоприемников» (под ред. В. В. Енютин, 1950).

Л. АКОПОВ

Учитель физики
школы № 43 Краснодара,
руководитель клуба

О. КЛЮЧНИКОВ

Инженер,
зам. руководителя
клуба «Электроник»

Четыре зайца студенческого отряда



Кажется, никого не надо уже убеждать, что программы для ЭВМ — это настоящая продукция, по сути своей такая же, как гайки, болты или, например, железнодорожные рельсы. А раз это продукция, то и производить ее лучше всего промышленным способом. Как на заводе.

Но постойте. Как производят на заводах гайки и болты, мы знаем. Там конвейер, там разделение труда, там стандарты, ОТК и госприемка. Неужели все это возможно при разработке компьютерных программ? Возможно! И даже необходимо.

Вот с какими идеями приступил к работе студенческий научно-производственный отряд МФТИ «Спектр». Ребята рассудили просто. Надо создать ССО, который вместо строительства коровников занимался бы «строительством» программ для персональных ЭВМ. При этом убивается сразу четыре зайца. Первый — появляются программы, столь необходимые сегодня стране. Второй — студенты работают

на ЭВМ, и при этом повышается качество их профессиональной подготовки. Третий — программы, распространяющиеся в школах, — это лучшая реклама родному вузу. Ну, и четвертый заяц — это прибавка к стипендии.

Отряд был организован в апреле 1986 г. Сначала работы велись на КУВТ «Ямаха», затем, с появлением в институте КУВТ-86, отряд начал писать программы для БК-0010. Написанные программы условно можно разделить на четыре класса.

Системные программы и инструментальные средства. Четвертый класс программ — это всевозможные графические редакторы, редакторы спрайтов, программа для составления видео-урока, а также «упаковщики» данных, ассемблеры, сеть для ПЭВМ «Микроша» и др.

Обучающие программы. Это в основном показанные «в разрезе» работающие физические приборы и устройства (лазер, двигатель внутреннего сгорания, атомный реактор, линза, поршень с газом) или игра в полет от Земли к Луне, в процессе которой играющий намертво запомнит первую и вторую космические скорости. (Об одной из таких обучающих программ, составленной «Спектром» совместно с К. М. Шоломием, журнал «Информатика и образование» подробно писал в № 3 за 1987 г.)

Третий тип программ — игры. Вообще-то написать игру не проблема. А вот придумать новую, свежую игру не так просто, а если еще надо, чтобы игра чему-то учила или тренировала, то задача становится совсем нетривиальной. СНПО МФТИ среди своих удачных работ в этой области считает игру

«Авиадиспетчер», в которой играющий превращается в диспетчера аэропорта и должен напряженно трудиться, чтобы продержаться смену без аварий. Ходит по Москве игра «Биомасса», в которой надо продумать стратегию, чтобы отгородиться со всех сторон от наступающей «биомассы». Чешскую игру «PEXESO» мы переложили на компьютерный язык и дополнили одним элементом — к каждой картинке дописали ее имя на иностранном языке. Пусть играющие заодно учатся.

Это серьезные программы, предназначенные для нашего брата — инженера-физика, инженера-математика. Спроектирован и воплощается пакет из 86 прикладных программ по курсам матанализа, матстатистики, алгебры, геометрии, теории вероятностей, уравнениям математической физики. Первые выпуски этого пакета продаются в магазине «Электроника» на Ленинском проспекте в Москве.

Помимо этого, отряд пишет программы, которые ни



в какие классификации не лезут. Куда, например, отнести программу, превращающую компьютер в шахматные часы или приставку к телефону? Мы называем их просто инициативными разработками.

Теперь о том, как организована работа. Отряд трудится в течение учебного года и один месяц в летние каникулы. Еженедельно проходят семинары СНПО, на которых обсуждаются рабочие вопросы. По вечерам бойцы отряда приходят в компьютерный класс, а некоторым машины выданы прямо в общежитие.

При помощи Калининградского молодежного творческого объединения «Импульс» отряд заключает договоры.

Там, где это возможно, отряд внедряет промышленные методы разработки программ. Например, идеи игр придумываются на семинарах, разработку алгоритма будущей игры ведет программист. Оформление поручается другому. Фирменную заставку к программе пришивает третий. Выделились и своеобразные «специальности» бойцов отряда. Например, специалист по ДВК, специалист по Т-языку, консультант по вычислительным методам и т. д.

Теперь о проблемах. Главная из них — техника. Писать хорошие программы для БК-0010 можно на машинах классом не ниже ДВК. А ДВК у нас всего один, и, хотя институт готов купить для нас машины, фондов на них не дает никто. Однако мы с оптимизмом (а как же иначе!) смотрим в будущее. В конце концов все проблемы решаются.

В. ФОМИН,
командир СНПО «Спектр»
МФТИ

Приставка-таймер к программируемому микрокалькулятору

Предлагаемая конструкция разработана и испытана в нашей школе применительно к МК «Электроника БЗ-34». Но она может работать вместе с любым программируемым микрокалькулятором. Это своего рода сервисное устройство, расширяющее его возможности.

Представьте себе, что ваш МК выполняет расчеты по большой программе с выводом промежуточных результатов. Как известно, для этого несколько раз в программу вводится команда С/П. Иногда расчеты по программе длятся десятки минут. Избавить вас от томительного ожидания поможет нежный прерывистый звук, он пригласит вас для считывания ответа. Этот сигнал появляется всегда по команде С/П при индикации промежуточных и конечного результатов. Если программа еще не закончена, то после определенной паузы, длительность которой устанавливается пользователем, устройство автоматически запускает МК для дальнейших расчетов.

Промышленность начала серийный выпуск демонстрационных калькуляторов для школы. Если с таким калькулятором совместить предлагаемую приставку, то учителю не нужно будет выходить к доске и после каждого останова программы нажимать клавишу С/П для дальнейших расчетов.

Например, при исполнении на МК программы по ранжированию таблицы после соответствующего фрагмента высветится первый МИН-ЭЛЕМЕНТ. Затем последует пауза, в течение которой учащиеся заполняют первую

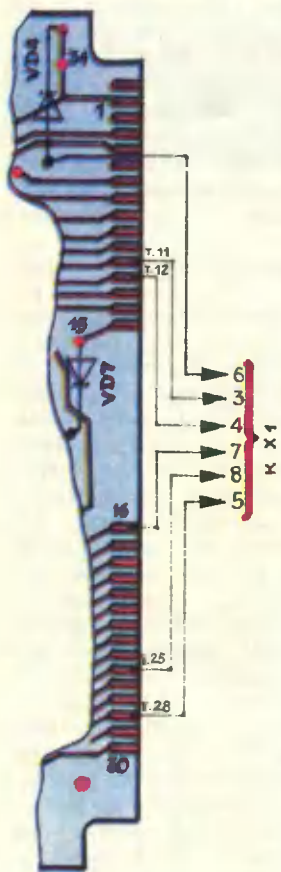
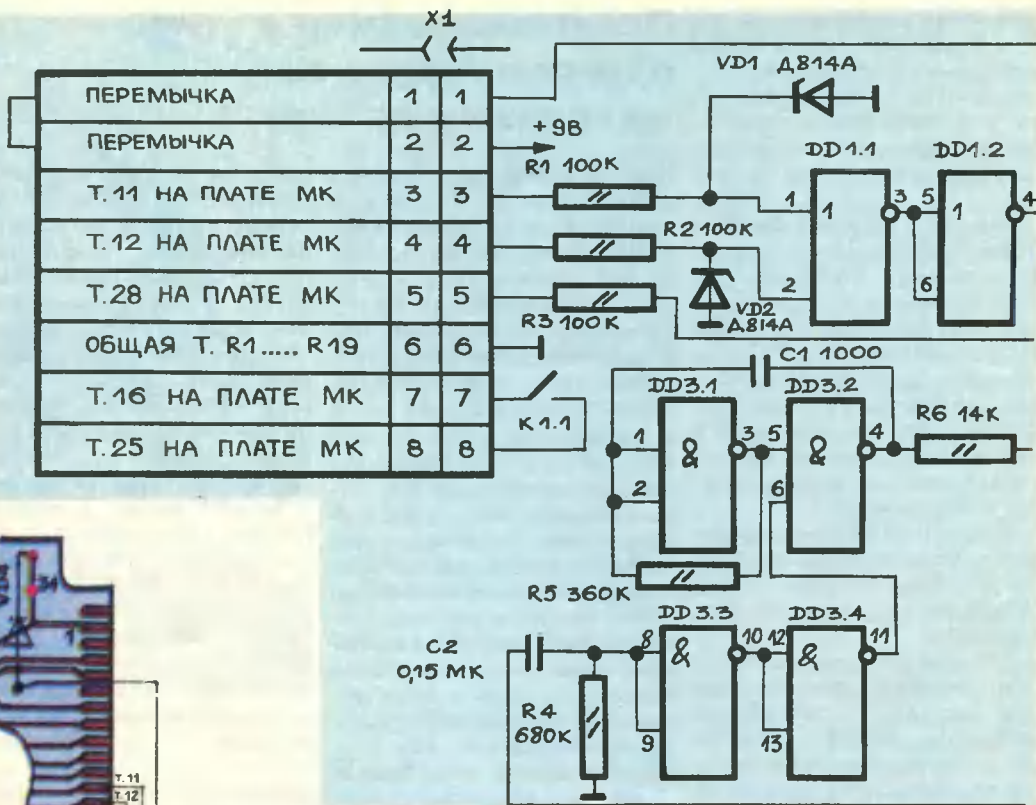
клеточку в заготовленной таблице. После этого МК автоматически запускается до индикации следующего МИН-ЭЛЕМЕНТа. Затем снова пауза... и т. д. до окончания ранжирования всей таблицы.

Кроме указанных функций устройство может играть роль таймера для отсчета различных промежутков времени (правда, с известной точностью). Для этого следует в МК ввести программу расчета какой-либо функции (например, $F \sin$), и организовать вычисления по циклу. Чем больше число циклов, тем больше времени требуется для расчетов. Необходимо заранее для данного МК опытным путем определить время, затраченное на

один цикл: $\tau = \frac{t_{\text{общ}}}{N}$ ($t_{\text{общ}}$ — общее время, N — число циклов).

В конце заданного промежутка времени в динамике появится прерывистый звуковой сигнал, громкость которого можно менять. В этом режиме устройство можно широко использовать для физического эксперимента — отсчета промежутков времени, включения различных механизмов на запрограммированное время (для этого необходим дополнительный триггер) и т. д.

Принцип действия устройства таков. С определенных точек печатной платы МК (они указаны на схеме) снимаются сигналы и через резисторы R_1 , R_2 и R_3 подаются на элементы DD1.1 и DD2.1, которые служат для формирования тактовых импульсов. Элементы DD1.2 и DD2.2 являются инверторами. С выхода 4 элемента DD2.2 импульсы положитель-



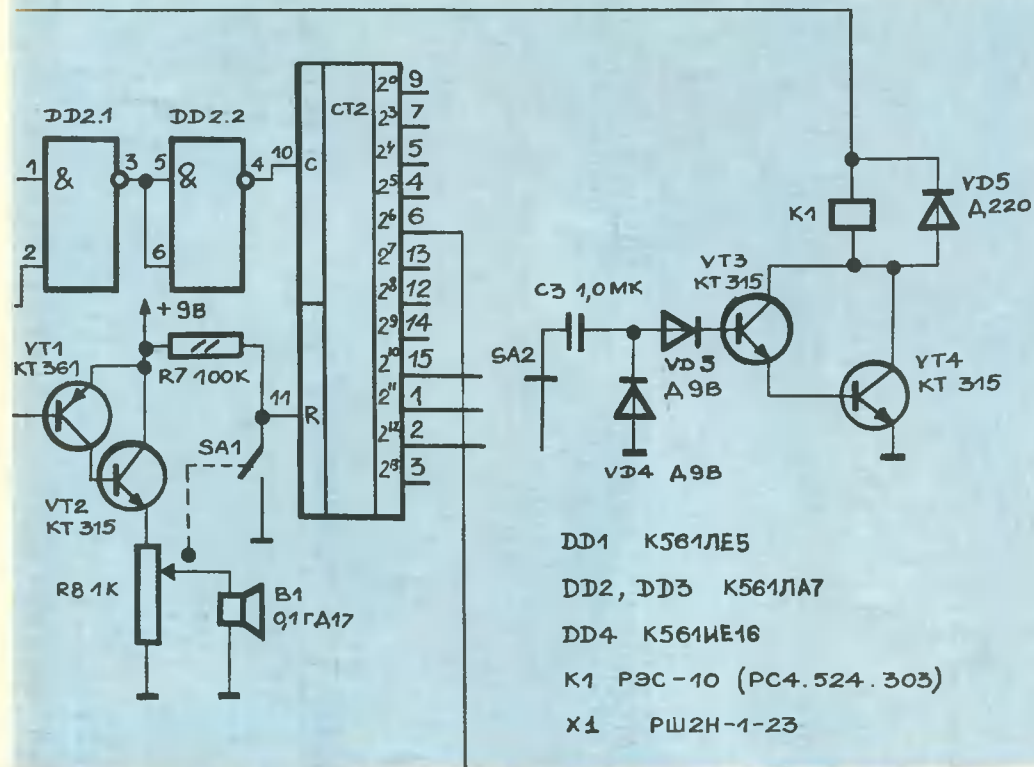
ной полярности поступают на вход 10 счетчика DD4. Частота этих импульсов резко изменяется в зависимости от режима работы МК. При исполнении программы частота

их — единицы герц, а при служит звуковой генератор индикации любого числа — десятки килгерц. Если произойдет останов программы (встретится команда С/П), то счетчик начнет быстро заполняться. Чем больший коэффициент пересчета соответствует данному выходу счетчика, тем больше времени пройдет до появления на этом выходе логической «1», тем больше пауза для считывания результатов. Длительность паузы устанавливается переключателем SA2. В момент появления высокого уровня элементы цепи C3, VD3, VD4 сформируют короткий импульс, открывающий ключ на составном транзисторе (VT3, VT4). Реле K1 срабатывает, и своими контактами, подключенными параллельно клавише С/П, вновь запускает МК.

При вводе программы или при необходимости выключить устройство нужно выключатель SA1 перевести в разомкнутое состояние. При запуске МК следует замкнуть контакты SA1 и установить резистором R8 требуемую громкость звука.

Устройство удобно собирать в корпусе транзисторного приемника (из наборов), используя имеющийся там динамик, а также регулятор громкости и выключатель питания (в качестве SA1). У нас же кон-

При формировании прерывистого звукового сигнала



струкция собрана в корпусе от громкоговорящей приставки к телефону.

Следует учесть, что если сопротивление звуковой катушки динамической головки окажется меньше 16 Ом, то последовательно с резистором R8 нужно включить резистор на 100 Ом.

Монтаж печатный. Приставка питается от батареи «Крона». Потребляемый ток незначителен. Отдельного выключателя питания нет. Его заменяют перемычки между контактами 1, 2 разъема XS1. Гнездовая часть разъема XS1, установленная на корпусе МК или спрятанная в один из отсеков для аккумуляторов, соединяется с указанными точками платы МК тонкими многожильными проводами. В качестве SA2 подойдет любой малогабаритный переключатель на 3 положения.

А. КАРАВАЕВ

Дарите детям компьютеры

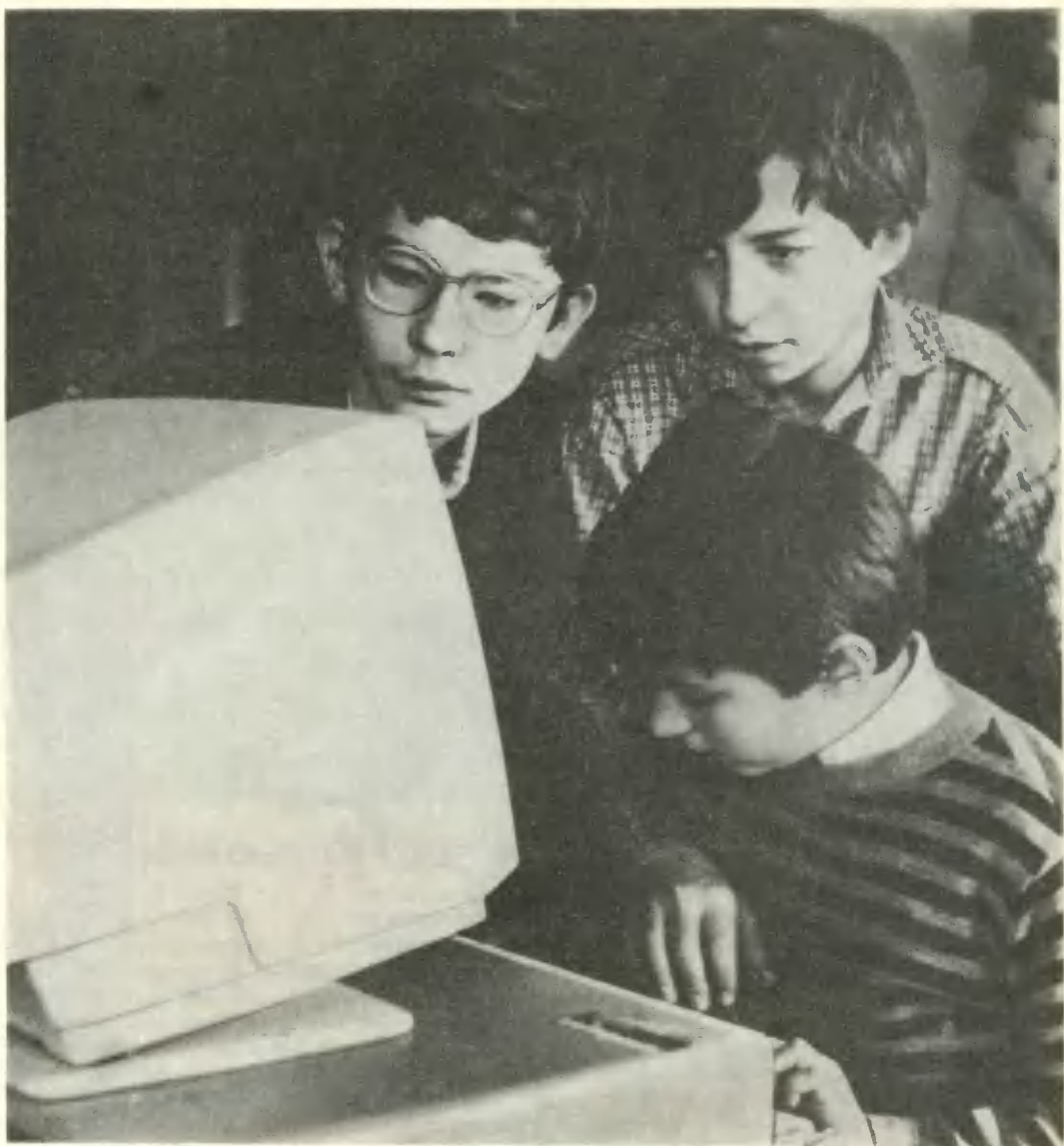
В сентябре в московском городском Дворце пионеров и школьников дарили подарки. Казалось бы, обычное дело: ведь, где дети, там забота, внимание взрослых. Дело — обычное, а вот подарки — редкие, да такие, о которых сегодня каждый школьник мечтает, — компьютеры.

Несколько лет во Дворце пионеров и школьников на Ленинских горах действует клуб космонавтики. У ребят отличный планетарий, настоящие тренажеры, есть выставка «атрибутики» космонавтов. Уже сегодня здесь можно отправиться в увлекательные «путешествия» по Вселенной. При этом выбор режима полета, управление системами «корабля» осуществляются с помощью «бортового компьютера»,

роль которого выполняет диалого-вычислительный комплекс ДВК-2М. Многие члены клуба неплохо освоили ЭВМ.

Юные космонавты дружат с учеными, сотрудниками Института космических исследований АН СССР.

Один из добрых и давних друзей клуба космонавтики — директор этого института лауреат Государственной премии СССР, академик Роальд Зиннурович Сагдеев. Не так давно Роальд Зиннурович по приглашению видного общественного деятеля США Леонарда Мотнера прочитал серию лекций в Калифорнийском университете. Гонорар академик попросил перевести в необычной форме: приобрести на эти деньги несколько современных компьютеров.



В сентябре 1987 г. наша страна, все человечество отмечало 30-летие начала космической эры. Москва принимала гостей-участников Всемирного форума, посвященного этому событию.

В один из дней группа советских и американских ученых и космонавтов посетила клуб космонавтики Дворца пионеров на Ленинских горах. Среди почетных гостей были академик Р. Сагдеев, Л. Мотнер, председатель совета директоров фирмы «Полароид» У. Маккьюн, испол-

нительный директор Института мировых проблем им. Д. Эйзенхауэра С. Эйзенхауэр и многие другие. Приехали летчик-космонавт СССР, Герой Советского Союза А. Серебров и астронавт США К. Салливан. Интересной и насыщенной была встреча будущих космонавтов с участниками форума. Они даже совершили учебный «космический полет». А центральным событием стала передача Р. Сагдеевым в дар клубу трех персональных компьютеров.

Компьютерный центр клуба с первых дней заработал с полной нагрузкой. Наладить его работу помогли сотрудники Института космических исследований Юрий Григорьевич Круглов и Виктор Владимирович Баркия.

Мы полностью поддерживаем инициативу академика Р. Сагдеева и сотрудников ИКИ по передаче компьютерной техники в подростковое любительское объединение. Кто следующий?

А. ПАРАМОНОВ

Сказка-ложь...

«Впервые в истории освоения Арктики на дрейфующей льдине работают математики». Это подзаголовок опубликованного «Комсомольской правдой» репортажа ее спецкора В. Юнисова со станции «Северный полюс-28». Репортаж называется «Алгоритмы океана».

Оставим на совести автора заголовок и некоторые столь же бессмысленные выражения («Труд полярного математика (есть теперь такая профессия!) утомителен»; «... скромные полярные математики в своих постоянных вахтах на ЭВМ, в математических лабиринтах... открыли новую страницу в истории освоения Арктики — компьютерную»). Разберем единственную фразу этого репортажа, раскрывающую существо применения ЭВМ.

«Язык машины», «фортран» или «алгол», превращает звуковые сигналы в обычные римские цифры, рассказывая, таким образом, о физических величинах скорости течений, солености и плотности воды, температуры воздуха».

Начнем с того, что, во-первых, *величины* температуры, плотности и т. д. могут быть только физическими. Так что слово «физические» здесь излишне.

Во-вторых, слово «величина» тоже излишне в этом контексте. Скорость, плотность и т. д. выражаются именно величинами. Однако на практике, и в частности при работе с ЭВМ, нам приходится иметь дело с числами, показывающими отношение данной величины к другой величине того же рода, принятой за

единицу. Это известно — или во всяком случае должно быть известно — каждому школьнику. В том числе и тому, который потом станет спецкором.

В-третьих, цифры, с которыми работает описываемая в репортаже ЭВМ СМ-1800, все же, вероятно, арабские. Вряд ли для работы в условиях Арктики в ЭВМ специально построили дополнительный блок, который, скажем, при тридцатишестиградусном морозе высвечивает на экране

**ФИЗИЧЕСКАЯ ВЕЛИЧИНА
ТЕМПЕРАТУРЫ ВОЗДУХА РАВНА
МИНУС XXXVI ГРАДУСОВ ЦЕЛЬСИЯ.**

И наконец, последнее. Представьте себе, что на СП-28 работает лаборант. Родом он из Бомбея, понимает лишь английский и хинди. На одном из этих языков ему дали инструкцию: каждые полчаса записывать в журнале температуру воздуха. Если продолжить репортаж в том же стиле, в нем может появиться фраза:

Язык лаборанта (т. е. английский или хинди) превращает показания термометра в обычные римские (или арабские?) цифры.

Будем надеяться, что В. Юнисову все же известно: Фортран и Алгол — языки, которые делают понятным для ЭВМ задания, сформулированные в программе. Но никакой язык не может превращать звуковые сигналы в цифры, не может язык и рассказывать, даже «таким образом».

В заключение констатируем, что ни Фортран, ни Алгол не смогут превратить дилетанта в человека, грамотно рассказывающего о возможностях современных компьютеров. Прежде чем писать об ЭВМ, полезно ознакомиться с началами информатики.

А. ХАЛАМАЙЗЕР

ИНФОРМАЦИЯ

Выставка для вас

В рамках мероприятий, проводимых на ВДНХ СССР в 1988 г. Минпросом СССР, состоится тематическая выставка «Электронно-вычислительная техника в народном образовании». Ее цель — показать особую роль электроники и микропроцессорной техники в овладении школьниками основами научных знаний, формировании эстетических вкусов. Там будут представлены фрагменты кабинетов вычислительной техники, натурные образцы ВТ и специального оборудования используемого в школах и учреждениях народного образования. Будет освещаться опыт союзных республик по внедрению ВТ в учебный процесс в школах и УПК.

На базе выставки пройдет всесоюзный семинар «Об итогах изучения курса «Основы информатики и вычислительной техники» и задачах на новый учебный год», на котором встретятся преподаватели и методисты учебно-воспитательных учреждений, научные работники, представители органов народного образования, печати.

Важная часть выставки — демонстрация пакетов прикладных программ. Принятые комиссией Минпроса СССР программные средства будут копироваться для организаций на их магнитные диски (при наличии писем-заказов).

К итогам конкурса

В. БЕЛОШАПКА

Канд. физ.-мат. наук

НИИ СиМО АПН СССР

Секретарь конкурсной комиссии

Конкурс на учебник информатики

Начнем с обзора основных событий этого конкурса. 10 апреля 1986 г. в «Учительской газете» была помещена заметка, в которой объявляется конкурс на создание учебника «Основы информатики и вычислительной техники». Была сформирована конкурсная комиссия из 36 человек, среди них — учителя школ, преподаватели техникумов, методисты, ученые, специалисты в области программирования, представители Минвуза и Госпрофобра. Председатель комиссии — академик А. П. Ершов. Вскоре Минпрос получил 24 заявки на участие от авторских коллективов из Челябинска, Симферополя, Шуи, Брянска, Киева, Николаева, Москвы и Ленинграда и других городов.

К 10 апреля 1987 г., через год после объявления конкурса, в Минпрос представлено 9 рукописей под девизами, а также конверты под теми же девизами со сведениями об авторах. В соответствии с условиями конкурса девизы с рукописей были изъяты и каждой из них был присвоен свой шифр. Затем рукописи переданы комиссии для рассмотрения. Конверты со сведениями об авторах остались запечатанными до конца конкурса и были вскрыты только после вынесения комиссией окончательного решения.

Первичное рассмотрение рукописей комиссией выявило несоответствие двух из них общим требованиям к школьным учебникам. Остальные семь при-

няты к дальнейшему рассмотрению. Для повышения степени объективности оценки рукописей было принято решение разослать их на экспертизу и рецензирование, был утвержден план рассылки. Параллельно с этой работой шло размножение рукописей для рассылки. Здесь уместно посвятить на серьезные трудности, которые порой возникали в нашей работе. Например, при организации тиражирования выяснилось, как неохотно принимают наш заказ типографии. Дело в том, что его параметры, а именно большой объем рукописи (до 600 машинописных страниц) и малый тираж (150 экземпляров) при имеющейся тарифной сетке для типографии совсем невыгодны. При этом финансовый статус нашего конкурса оставался довольно неопределенным. Смета конкурса была утверждена как раз к моменту его завершения. С большими трудностями пришлось столкнуться в решении вопроса о размещении в гостинице членов комиссии. Но, впрочем, трудности были преодолены и план рассылки в целом выполнен. В рецензировании рукописей приняли участие 13 научных и педагогических организаций, 21 специалист, представивший индивидуальные рецензии. Было получено 44 акта экспертизы и 46 индивидуальных рецензий, из которых 22 от учителей.

В июле 1987 г. после серьезной ра-

боты над рукописями комиссия соби-
ралась на последнее заседание. Про-
ходило обсуждение каждой рукописи,
изучение полученных актов экспертизы
и рецензий. Был принят следующий
критерий присуждения мест:

первое место присуждается рукописи,
которая удовлетворяет в достаточной
степени школьным учебникам, и после
незначительной редакторской правки
может быть рекомендована к педагогиче-
скому эксперименту в качестве учеб-
ника информатики для средней обще-
образовательной школы;

второе место присуждается рукописи,
которая удовлетворяет не в полной мере
требованиям к школьным учебникам
и после определенной доработки может
быть рекомендована к педагогическому
эксперименту в качестве учебника ин-
форматики для средней общеобразо-
вательной школы;

третье место присуждается рукописи,
которая не удовлетворяет требованиям
к школьным учебникам, но может быть
рекомендована для использования в
средних учебных заведениях в качестве
дополнительной учебно-методической
литературы по информатике.

В результате длительной серии туров
тайного голосования комиссия вынесла
свое решение. На следующий день
были вскрыты конверты со сведения-
ми об авторах и принято следующее ре-
шение:

1. Первое место не присуждается
ни одной рукописи, представленной на
конкурс.

2. Второе место присуждается руко-
писи под шифром КИ-06(10—11).

Рекомендуется к награждению второй
премией авторский коллектив в со-
ставе: *Каймина В. А.*, доцента кафедры
кибернетики МИЭМа; *Щеголева А. Г.*,
инженера кафедры кибернетики МИЭМа;
Гиглавого А. В., старшего научного со-
трудника Института электронных управ-
ляющих машин; *Ерохиной Е. А.*, сту-
дентки V курса факультета приклад-
ной математики МИЭМа; *Федюши-
на Д. П.*, директора компьютерной физ-
матшколы МИЭМа.

3. Третье место присуждается двум
рукописям под шифрами КИ-0 (10—11)
и КИ-04(10—11).

Рекомендуются к награждению
третьими премиями авторы: *Вер-
лань А. Ф.*, профессор Киевского поли-
технического института; *Касаткин В. Н.*,
доцент Симферопольского государст-
венного университета (первая ру-
копись);

Автор *Вершинин О. Е.*, доцент кафе-
дры микроэлектроники и технологии
Ленинградского электротехнического
института. (вторая рукопись).

После доработки и тщательного
научного редактирования рукопись
КИ-02(10—11) может быть рекомендо-
вана к изданию в качестве дополни-
тельного методического материала
по курсу информатики. Рукопись
КИ-04(10—11) после соответствующей
переработки может быть рекомендована
к изданию в качестве книги для чте-
ния по информатике.

Для того чтобы обсудить результа-
ты голосования необходимо хотя бы бег-
ло провести обзор представленных ра-
бот. Итак, рукопись КИ-02(10—11). Дё-
виз — «Начало». Авторский коллек-
тив: *Верлань Анатолий Федорович* —
доктор технических наук, профессор
Киевского политехнического институ-
та, *Касаткин Валентин Николаевич* —
кандидат педагогических наук, доцент
Симферопольского государственного
университета.

Вместе с рукописью учебника на
конкурс были представлены следующие
материалы: приложения, содержа-
щие алгоритмы, примеры и программы;
сборник упражнений и задач по осно-
вам программирования на Бейсике;
методические указания для учителей;
пояснительная записка; распределение
учебного материала; перечень учебно-
демонстрационного и программного
обеспечения. Суммарный объем пере-
численных материалов — 569 маши-
нописных страниц, из них 364 — учеб-
ник с приложениями. Таким образом,
на конкурс представлен не просто
учебник, а солидный учебно-методиче-
ский комплект. Материал разделен на
параграфы, содержание каждого из ко-
торых предлагается для усвоения на
одном уроке. В конце каждого пара-
графа имеются вопросы и упражне-
ния для самостоятельной работы уча-

щихся. Язык рукописи в основном прост и доступен для учащихся. Несомненно, методический аспект данной рукописи проработан солидно и основательно. Но, к сожалению, этого нельзя сказать о научно-содержательном аспекте. Научная концепция, лежащая в основе учебника, аморфна и архаична. Отсутствует ясное понимание задач данного конкурса. Имеются диспропорции в объемах учебных тем. Например, в пяти параграфах разбираются различные системы счисления, на 150 страницах изучается MSX — Бейсик, а вопросам постановки задач для решения на ЭВМ и моделирования уделена пара страниц. Текст изобилует неточностями, некорректным использованием терминов. В рукописи не отражен межпредметный характер данной дисциплины. Примеры и упражнения берутся, как правило, из математики. Материал излагается скучно. Все сказанное резко снижает ценность данной рукописи.

Рукопись КИ-03(10—11). Девиз — «Дорогу осилит идущий». Автор — Федотов Валерий Павлович, кандидат физико-математических наук, доцент кафедры математики Шуйского государственного педагогического института им. Д. А. Фурманова.

Работа написана на основе ясной, целостной, последовательной, корректной авторской концепции. Автор смотрит на информатику с широких общенаучных позиций. Это позволяет ему, например, при рассмотрении вопросов алгоритмизации использовать лингвистические аналогии. Такая позиция сообщает обучению ценный межпредметный, в частности гуманитарный, характер. Изложение материала живое и доступное. Однако методическая проработка в рукописи практически отсутствует. Нет ни задач, ни вопросов. Нет вычленения основных знаний и умений. Алгоритмическая линия дается лишь в теоретическом аспекте. Нет обучения составлению алгоритмов. Уровень учебника недоступен, по-видимому, для большинства школьников. Обращая внимание читателя на девиз рукописи, позволим себе задать вопрос: «Действительно ли идущий осилит эту доро-

гу?» Имеют место неоправданные диспропорции в объемах учебных тем. Например, системам счисления отведена целая глава — пятнадцать (!) параграфов. Эти недочеты не позволяют, к сожалению, рассматривать данную рукопись в качестве школьного учебника информатики.

Рукопись КИ-04(10—11). Девиз — «Лед и пламень». Автор — Вершинин Олег Емельянович, кандидат технических наук, доцент кафедры микроэлектроники технологии радиоаппаратуры Ленинградского орденов Ленина и Октябрьской Революции электротехнического института им. В. И. Ульянова (Ленина).

Данная рукопись содержит 464 машинописные страницы. Если отвлечься от того контекста, в котором рассматриваются рукописи (конкурс на школьный учебник информатики), то предлагаемая книга производит благоприятное впечатление. Этому способствует широкая эрудиция автора, хорошее владение материалом (неточности встречаются довольно редко), увлекательно-повествовательный стиль изложения. На страницах рукописи есть исторические обзоры, обсуждение вопросов устройства и архитектуры ЭВМ, принципы работы и технологии производства интегральных схем, вопросы функционирования автоматизированных систем управления и связанных с этим математических моделей (системы массового обслуживания, имитационные и игровые модели) и многое другое. Однако, восстановив этот контекст, мы получим иную картину. Повествовательность изложения оборачивается отсутствием учебной направленности, широта спектра рассматриваемых вопросов — невозможностью удовлетворительного усвоения каждого из них. Причем положение не спасают и имеющиеся в конце некоторых параграфов контрольные вопросы. Такое сочетание, как Алмир и Бейсик, вызывает по меньшей мере удивление. Да и вообще, алгоритмическая линия теряется в этом море сведений. Оценка данной рукописи существенно зависит от того жанрового контекста, в котором мы ее будем рассматривать.

Рукопись КИ-05(10—11). Девиз — «Кажное... сочинение надо уметь изложить доступно. Кант». Автор — Копылов Георгий Николаевич, кандидат физико-математических наук, доцент кафедры математического анализа Брянского педагогического института.

Рукопись содержит 211 страниц. По видимому, основная задача, которую поставил перед собой автор данной рукописи, — сделать изложение учебного материала простым и доступным. Об этом свидетельствует и девиз, под которым рукопись пришла на конкурс. Простоту и доступность следует отнести к несомненным достоинствам предложенного учебника: изложение ведется живо, образно, конкретно и доходчиво. Однако при этом следует отметить ряд недочетов. Во-первых, это касается авторского взгляда на информатику. Во введении после предварительного обсуждения приводится такая формулировка: «Информатика — наука об электронно-вычислительных машинах». Далее следует отметить неточности и мелкие ошибки. Конкретность подхода при ориентации на микрокалькулятор и ДВК оборачивается негативной стороной. Следует отметить, что данная рукопись нацелена в основном на одну группу учебных задач, а именно на отработку практических навыков работы с компьютером. Остальные же проработаны слабо.

Рукопись КИ-07(10—11). Девиз — «Системность». Автор — Чистяков Василий Юрьевич, кандидат химических наук, заведующий кафедрой информатики и вычислительной техники Челябинского ордена «Знак Почета» государственного педагогического института.

Данная рукопись (366 страниц) интересна как попытка реализовать в школьном курсе методологически широкие позиции, сообщить предмету тот междисциплинарный и системный характер, который присущ информатике в современном естествознании. Это авторская позиция отражена в девизе рукописи. И действительно, широта спектра рассматриваемых вопросов необычайна. Автор с легкостью перехо-

дит от обсуждения вопросов принципиальных и философских и таким техническим деталям, как указание конкретных ГОСТов или системы команд микрокалькулятора МК-56. Но, к сожалению, эта многоуровневая конструкция страдает от неточностей, ошибок, некорректного использования терминов, просто нелепостей. Общие позиции непоследовательны и также профессионально некорректны. К этому следует добавить, что, несмотря на наличие контрольных вопросов, рукопись не имеет учебной направленности. Язык изложения путаный и наукообразный, терминология непродуманная. Все это позволяет констатировать: данную попытку построения курса информатики следует признать неудачной.

Рукопись КИ-09(10—11). Девиз — «Структурал». Авторский коллектив: Прядко Владимир Алексеевич, кандидат технических наук, доцент, заведующий кафедрой математики Николаевского государственного педагогического института; Белый Юрий Александрович, кандидат физико-математических наук, доцент кафедры математики Николаевского пединститута; Гершаник Владимир Ильич, кандидат технических наук, доцент кафедры математики Николаевского пединститута.

Особенность рукописи (278 страниц) — систематическое использование при изучении алгоритмизации и программирования специально разработанного авторами алгоритмического языка Структурал (структурный, алгоритмический). Этот язык, в соответствии с названием, является языком структурного типа. Основная идея авторов при его разработке состояла в совмещении на базе данного языка изучения алгоритмизации и программирования. Однако, не вдаваясь в детали, отметим, что сравнение Структурала с алгоритмическим языком действующего учебного пособия, особенно в последней его версии (объединенное учебное пособие «Информатика 9—10»), проходит не в пользу первого. Структурал еще далек от реализации, а алгоритмический язык уже реализован на ряде учебных компьютеров (Е-практикум). Отметим также, что в целом

рукопись носит программистский характер и слабо проработана в методическом отношении.

Рукопись КИ-06(10—11). Девиз — «Что и требовалось доказать». Авторский коллектив: Каймин Виталий Адольфович; кандидат технических наук, доцент кафедры кибернетики Московского института электронного машиностроения; Щеголев Александр Гелиевич, инженер кафедры кибернетики МИЭМа; Гиглавый Александр Владимирович, кандидат технических наук, ст. науч. сотр. Института электронных управляющих машин; Ерохина Елена Альфредовна, студентка V курса факультета прикладной математики МИЭМа; Федюшин Дмитрий Петрович, директор компьютерной физматшколы МИЭМа.

102 Вместе с текстом учебника представлены: терминологический словарь (46 терминов); сборник упражнений и задач; программа, положенная в основу данного курса; объяснительные записки по курсу и практикуму; перечень учебных программных средств, которые предлагается использовать на практикуме (353 машинописные страницы). Данная рукопись заняла второе, самое высокое в данном конкурсе место, поэтому будет резонно рассмотреть ее несколько более подробно.

Учебник написан в соответствии с конкурсной программой, при этом программа не оказывает довлеющего воздействия, авторы проводят в тексте свое понимание информатики. Изложение современно, учитываются актуальные тенденции, привлекаются новые понятия и методы. Есть определенная методическая проработка, как это явствует из списка приложений, а также отметим, что после некоторых параграфов имеются контрольные вопросы. Список программного обеспечения и объяснительная записка к нему позволяют предположить, что вопросам программной поддержки курса уделяется большое внимание. Ряд положений, сформулированных в объяснительной записке, например то, что правила управления учебными программными средствами должны не меняться от программы к программе, а быть унифицированными, заслуживают одобре-

ния. К тому же авторы утверждают, что располагают прототипами и эскизными образцами приведенных в списке программных средств, что свидетельствует в пользу технической реализуемости этих программ.

Однако у учебника имеется и ряд серьезных недостатков. Вот некоторые.

При введении понятия «информация» допускается глубокое смещение содержательного и формального, объективного и субъективного его аспектов, не происходит выявления из общеразговорного смысла этого слова тех принципиально важных моментов, которые дают наполнение соответствующему общенаучному термину. Не отмечен объективный и формализованный характер его смыслового содержания. В итоге научная и дидактическая ценность введенного термина становится сомнительной. Одним из проявлений такого взгляда на вещи (а речь идет не столько о термине, сколько об авторской позиции) является систематическое смещение в рукописи таких понятий, как «сведения», «знания», «данные», «информация». Другой иллюстрацией терминологической некорректности служит помещенное в словаре объяснение термина «модель математическая», там написано, что это «система математических соотношений, отображающих структуру объекта». Заметьте, не объект или его существенные свойства, а «структуру объекта», таким образом, модель — это модель модели. Такая двусмысленность и неотработанность терминологии, безусловно, не способствуют достижению учебных целей.

Другая особенность рассматриваемой рукописи — декларация всеобщего характера многих ее положений, которые по своему статусу таковыми не являются. В тексте сказано: «Основой всякого языка является алфавит». Однако чуть выше языком объявляется язык жестов, а в другом месте язык зрительных образов. А между тем это языки иероглифического типа. Еще пример, из текста: «Точные и математически строгие постановки задач — ключ к успеху в любой деятельности...»

На следующем примере остановимся подробнее. В учебнике излагается некоторая технология решения задач с помощью ЭВМ. И если отнести к ней как к таковой, имея в виду, что она относится лишь к некоторому классу таких задач, то ее изложение можно отнести к достоинствам рукописи. Но в рукописи после критического рассмотрения некоторого неправильно-го способа написано: «Мы будем рассматривать другой, действительно общий способ составления алгоритмов и решения задач на ЭВМ, состоящий из следующих этапов». И далее идет список из семи пунктов. Причем записан он с использованием алгоритмической нотации, что, видимо, соответствует представлениям об алгоритмическом его характере.

Но обратимся хотя бы ко второму его пункту, который звучит так: «Определение метода решений». Что такое метод, хорошо написано в другом месте рукописи: «общий способ действий, дающий решения для класса задач». Приведенный пункт показывает, что излагаемый способ имеет отношение к задачам инженерного типа, к таким, относительно которых известно, как, в принципе, они делаются, т. е. известен метод, который нужно применить в некоторой конкретной ситуации. Приведенный пункт не имеет отношения к задачам исследовательского характера, где общий способ часто возникает как обобщение серии частных результатов. И не надо думать, что задачи должны быть очень несложными. Школьник решая уравнение $x^2 - 2x + 3 = 0$, если он еще не знает общей формулы, находится именно в такой ситуации. Еще одно ограничение общности принятого подхода связано с тем, что, как показывает опыт, такая формализация процесса решения неплохо действует в математических классах, а в обычных это приводит к быстрой потере интереса к предмету.

В четырех параграфах первой главы данной рукописи (10 страниц) излагаются вопросы логики. Как известно, бурное развитие математической логики в XX веке оказало серьез-

ное влияние на становление информатики. Поэтому сам факт появления на страницах учебника информатики таких параграфов заслуживает внимания. К сожалению, взгляды авторов на этот предмет глубоко эклектичны. Здесь происходит довольно болезненное наложение четырех позиций: формальной логики в смысле Аристотеля (суждения, утверждения, рассуждения и пр.), математической логики (логические связи и кванторы), логики как науки, изучающей законы мышления, и логики житейской.

Такая позиция неминуемо приводит к недоразумениям и ошибкам. Во-первых, если бы законы мышления совпадали с законами формальной логики, то не стояла бы задача формирования логического стиля мышления, на решение которой, по-видимому, направлены данные параграфы. Во-вторых, на страницах рукописи неоднократно в разных формах утверждается, что всякое (опять квантор всеобщности) суждение или утверждение является либо истинным, либо ложным. При этом тут же эксперимент предлагается как способ проверки истинности. А как быть, если получены данные как за, так и против? Но это еще полбеда. Рассмотрим следующее высказывание: «Я лгу». Приписывание ему как статуса истинного, так и статуса ложного за один шаг приводит нас к противоречию. В рукописи есть замечание, из которого явствует, что авторы что-то слышали о таких трудностях, но содержание этого замечания очень далеко от сути дела. А речь идет ни много ни мало, как о том феномене автореферентности, который последние восемьдесят лет находится в центре внимания логиков.

Теперь давайте рассуждать логически. Если авторы знали об этом, то, приняв столь некорректную позицию в учебнике, они поступили плохо, но если они не знали об этом, то они поступили еще хуже, ибо в этом случае им не следовало бы включать в текст рукописи изложение вопросов, в которых они не разбираются. Далее, имеется трудность методиче-

ского плана. Известно, что логика — дисциплина обобщающего характера, причем сфера, предоставляющая материал для этого обобщения, может быть разной, при этом можно получать разные логики, в частности отличные от принятой в рукописи. Если же мы хотим получить какой-то вариант формальной логики, то эта сфера должна быть довольно формализованной. Например, это могут быть математика, юриспруденция или программирование. Обучение логике до знакомства с такой сферой и без обращения к ней обречено на неудачу. Поэтому можно было бы понять появление этого раздела после основательной работы по обучению алгоритмизации и программированию или же в контексте школьной математики, но в рукописи материал дается практически конспективно.

С вопросом об изложении в рукописи разделов по логике тесно связаны разделы по проверке правильности алгоритмов и программ. В объяснительной записке содержание этих разделов объявляется «главной научно-методической целью курса», что и явилось основанием для включения в курс параграфов по логике. Здесь надо заметить следующее. Действительно, для овладения техникой доказательства программ нужна логическая подготовка. Причем ее уровень должен быть порядком выше, чем просто для составления тех же алгоритмов и программ и их отладки. И если во втором случае отработка логики может идти параллельно изучению алгоритмизации, то в первом (для доказательного программирования) необходимо довольно высокий начальный ее уровень. По нашему мнению, соответствующие разделы рукописи такой подготовки не дают. Далее для проведения процедуры доказательства необходимо построение формального описания программы. В теории доказательного программирования такой объект называется схемой программы. В курсе на роль такого объекта претендуют планы решения и описания методов, однако изложение способов их формального описания в тексте отсутствует. Если учитывать по

отношению к процедуре доказательства неформальную точку зрения, то успешное проведение процедуры не гарантирует отсутствия в программе ошибок, и такая процедура лишается статуса доказательства. Что же касается самой техники доказательства, то в тексте имеется несколько простых примеров, а в объяснительной записке отмечается: «Техника анализа результатов выполнения алгоритмов в различных ситуациях обобщается до исчерпывающего анализа...» Непонятно, где и когда произошло это обобщение. Далее, все известные процедуры доказательства проходят для программ определенных классов, с теми или иными ограничениями. Такого рода ограничения в рукописи не приводятся.

После рассмотрения нескольких простых примеров вопрос о доказательстве сложных алгоритмов решается так: «Анализ сложных алгоритмов, представляющих описание способов решения совокупности взаимосвязанных подзадач, строится на выделении вспомогательных алгоритмов и явном выделении постановок решаемых подзадач. Доказательство правильности строится на выделении вспомогательных утверждений и доказательств этих утверждений по отдельности». После прочтения этого абзаца, как и всей рукописи, неясно, происходит ли сужение класса алгоритмов. Мы склонны считать, что всякая задача может быть рассмотрена как совокупность взаимосвязанных подзадач. Не ясно, есть ли какие-то ограничения на структуру связей между этими подзадачами. Не ясно, как, авторы предлагают осуществлять это выделение подзадач?

Данная тема предстает нам недоступной для изучения в средней общеобразовательной школе; изложение этой темы в рукописи нельзя признать удовлетворительным.

В тесной связи со сказанным выше (логика и доказательство программ) находится и содержание главы «Основания информатики». Само это словосочетание звучит так же, как, например, «основания геологии». Информатика не является дедуктивной наукой, коей является математика. Авторы объяв-

ляют доказательное программирование ключевым для информатики вопросом. Этот тезис, превращая информатику в раздел дискретной математики, действительно придает ей дедуктивный характер. Но такой взгляд представляется нам узким и неплодотворным в научном и общеобразовательном отношении. Центр тяжести информатики не в работе с формальными объектами, что является прерогативой математики, а в самой формализации. В конце данной главы приводятся три закона формальной логики Аристотеля, и закон достаточного основания Лейбница. Они объявляются основными законами информатики. Такой поворот событий вызывает недоумение. Если эти положения предлагаются в качестве законов, то следовало бы дать такому их статусу хотя бы какую-то аргументацию. Но они помещены вне связи с основным текстом рукописи. Те краткие пояснения, которые даны вместе с формулировками, показывают, что авторы понимают их как некие рекомендации по преобразованию действительности.

Необходимо отметить, что обучение составлению алгоритмов в рукописи поставлено слабо. Соответствующие разделы производят впечатление комментариев. К чему? Возможно, к пробному учебному пособию под редакцией А. П. Ершова и В. М. Монахова. Отсутствует учебная направленность, изложение страдает несистематичностью. Например, беглое объяснение оператора присваивания появляется страниц через пятьдесят после начала его использования. Конструкции языка вводятся туманно и не всегда корректно. Из алгоритмического языка изъяты вспомогательные алгоритмы, что делает его ущербным. При этом без объяснения можно встретить записи типа « $X := \text{случайное}(1,100)$ ».

Вместе с уже отмеченной несистематичностью есть и неоправданная наукообразность языка, которая не окупается точностью формулировок, смысл некоторых фраз трудно понять, используются техницизмы. Исторический обзор содержит ряд неточностей. Раздел по применению ЭВМ не содержит кон-

ретного материала, носит поверхностно-футурологический, а порой и карикатурный характер. По содержанию и стилю все это очень похоже на рекламный проспект западной фирмы.

В рукописи не раскрыт межпредметный характер данного учебного предмета и интегрирующий характер соответствующей учебной дисциплины. Подавляющее большинство примеров — из математики. Гуманитарный аспект содержания предмета остался в тени. Можно сказать даже резче: рукопись, в частности ее язык, имеет выраженную антигуманитарную направленность. По-видимому, здесь, как, впрочем, и в других, уже отмеченных недочетах, можно в качестве причины указать резкую несбалансированность авторского коллектива. Нетрудно заметить, что почти все авторы имеют техническое образование и тот педагогический опыт, на который они опираются, — опыт преподавания в техническом вузе. И если придать авторской позиции ее естественный контекст, то все становится на свои места. Но средняя общеобразовательная школа — сфера иная. И для обучения школьников необходима очень серьезная переработка рукописи в научном и методическом отношении.

Обобщая, отметим, что задача, стоящая перед авторами, по своему объему, по-видимому не меньше объема уже выполненной работы. А создание стабильного учебника действительно ответственное дело. И можно надеяться, что авторы приложат необходимые в этом деле усилия.

Теперь перед нами все представленные на конкурс рукописи. Из предложенного обзора видно, насколько они разнообразны и трудносоотносимы. Среди множества параметров, по которым можно оценивать учебники, отметим несколько наиболее существенных:

содержание учебника должно быть адекватно стоящей перед данным предметом системе учебных целей;

изложение должно быть научно корректным;

изложение должно быть доступно учащимся;

содержание учебника не должно представлять собрание разрозненных сведений, т. е. изложение должно быть системным. Есть и другие существенные для оценки учебника положения. Но и этого довольно, чтобы убедиться в том, насколько трудно написать учебник. Возьмем, например, пару научность — доступность: если рукопись № 2 хорошо проходит по критерию доступности и плохо по критерию научности, то рукопись № 3 демонстрирует обратное. Но вот решение комиссии перед нами. Итак, победа в конкурсе за рукописью № 6, несмотря на те серьезные возражения, которые были во время предварительных обсуждений высказаны в адрес этой рукописи. Видимо, эта рукопись оказалась в целом более сбалансированной по отношению ко всей совокупности требований к учебнику, чем остальные.

Что еще дал нам конкурс кроме семи упомянутых рукописей? Новых авторов. Конкурс позволил многим из них продемонстрировать хорошее владение тем или иным видом учебно-методического жанра. В решении комиссии особо отмечены рукописи № 3 (В. П. Федотов) и № 5 (Г. Н. Копылов) фрагменты из них рекомендованы к публикации в журнале «Информатика и образование». Возможны и другие формы их использования.

Но учебника, настоящего учебника все же нет. Почему? Этот вопрос в ходе работы комиссии обсуждался не раз. Хотя полного единодушия на этот счет не было, позволим себе обратиться внимание читателя на два обстоятельства, каждое из которых делает появление действительно хорошего учебника почти невозможным.

Первое из них связано с той не вполне благополучной ситуацией, которая сложилась в отношении конкурсной программы. Как известно, она была подвергнута серьезной критике, в результате чего ее авторитет поколебался. Сама эта ситуация — отражение множественности точек зрения на задачи школьной информатики. И дело не просто в расхождении мнений по вопросу о той или иной программе, а в том, что, по существу, спектр запро-

сов общества к данному учебному предмету столь широк, что приведение этого многообразия к общему знаменателю, в единой программе и едином курсе происходит весьма болезненно. И это неизбежно при существующей унифицированной системе образования. Если же подойти к этому вопросу дифференцированно, т. е. строить обучение информатике, ориентируясь на те или иные задачи и контингенты учащихся, то проблема разработки программы и создания учебника получает более правильную и посильную постановку. В частности, почти каждая рукопись из представленных на конкурс имеет свою общеобразовательную специфику. Например, рукопись № 5 — это облегченный курс ознакомительного типа, № 3 — и по содержанию и по стилю изложения является гуманитарно ориентированным, № 6 — демонстрирует инженерный подход к информатике, № 9 — программистский.

Условия проведения данного конкурса были чрезвычайно жесткими. Для создания учебника даже при созревшем замысле необходимо реализовать настолько большой объем работ, что провести его в, так сказать, «факультативном» порядке, без освобождения автора от его основной работы, и в сжатые сроки (1 год) очень трудно. При таких жестких условиях из участия были авторские коллективы, от которых можно было ждать неплохих результатов.

Потребность школы в хороших учебниках остается. Поэтому можно надеяться, что первый конкурс такого рода не станет последним. И каждый просчет, выявленный здесь, — это урок, и его нужно учесть. В период работы комиссии вопрос о форме проведения конкурса обсуждался, и немало сторонников получило предложение, суть которого в том, что конкурс проводится в два этапа, причем на первом этапе проводится конкурс проектов учебников. Для адекватной оценки авторского замысла и возможности его реализации представленный проект должен включать в себя план-проспект, т. е. изложение идеи построения курса и описание его основных линий, проб-

ные главы и, возможно, список программного обеспечения. Для авторов, чьи проекты будут признаны лучшими, на втором этапе конкурса предоставляется возможность формирования авторского коллектива с обеспечением его работы в льготных условиях, т. е. при полном или почти полном освобождении от основной нагрузки. При этом возможны текущий контроль и коррекция со стороны экспертной комиссии. После подготовки рукописей комиссия дает оценку и рекомендации по их использованию.

Возможно, перед объявлением нового конкурса имеет смысл провести серьезный анализ таких основополагающих для курса вопросов, как его основные цели, а также его возрастного формата, и зафиксировать результаты этого анализа условиями конкурса. В отношении возрастного формата курса отметим следующее. В соответствии с представлениями, которые признаны в психологии вполне устоявшимися, опе-

рациональные возможности ребенка созревают раньше интеллектуальных. Поэтому представляется разумным следующую версию (такой взгляд на вопросы возрастного формата курса впервые мы встретили у Я. Э. Гольца в его докладе на Всесоюзной школе молодых ученых в Тбилиси, октябрь 1986 г.).

Младшие классы (границу должны установить физиологи) — небольшой, но концентрированный, практически ориентированный курс компьютерной грамотности (10—12 часов за 2 недели): практические навыки работы с готовым обеспечением плюс минимум теоретических сведений (немного об алгоритмах, компьютерах и программах). Такой курс может быть вкраплением в какой-либо другой предмет;

средние классы — использование учебных программных средств в других предметах;

старшие классы (по-видимому, не раньше восьмого) — хороший теоретический курс с общеобразовательным и развивающим потенциалом.

А. ГЕГЕЧКОРИ

Ретроспективные замечания о конкурсе учебников

Итак, конкурс завершен. По единодушному мнению комиссии, первая премия никому не присуждена. Иными словами, ни одна из семи представленных на конкурс рукописей не может рассматриваться как готовый учебник, по которому можно учиться в школе.

Вторую премию после дополнительного тура голосования большинством в один голос получила рукопись коллектива авторов под руководством В. А. Каймина. В протоколе конкурсной комиссии отмечено, что после значительных доработок она может претендовать на право называться школьным учебником.

Что же явилось причиной столь печального итога конкурса? Думаю, было бы наивным объяснять это некомпетентностью авторов рукописей. Дело не в этом или по крайней мере в большин-

стве случаев далеко не только в этом.

Есть целый ряд объективных причин, и хочется надеяться, что Минпрос СССР постарается устранить их при проведении очередного конкурса.

Первая — программа курса. По ее поводу было высказано много разнообразных критических замечаний. Не считая себя компетентным рассуждать на тему о том, насколько они справедливы, хочу лишь заметить, что этот вопрос надо было бы рассмотреть тщательнее. Возможно, конкурсу учебников следовало бы предварить конкурс программ (или идей, называйте как хотите), из которых было бы видно, чему и как авторы собираются обучать. Пока получилось, что участники конкурса оказались скованы рамками программы, с которой, возможно, и не были согласны. Кто знает,

сколько хороших идей было из-за этого потеряно.

Вторая — время. За отведенный срок написать учебник было, конечно же, очень сложно. Предложенный темп оказался слишком высоким. Многие авторы, не выдержав его, выбыли из состязания; те же, кто с честью вышел из борьбы с жестоким цейтнотом, не убергли свои рукописи от откровенных «ляпов», которые вряд ли допустили бы, работая спокойно. Буквально в каждой рукописи можно было выделить разделы, писавшиеся впопыхах.

Третья — загруженность участников конкурса другими делами. Никто их от основной деятельности не освобождал, и работать над рукописями им приходилось, как говорится, «в свободное вре-

мя». Создать хороший учебник очень трудно, думаю, было бы целесообразно предоставить нескольким авторским коллективам творческие отпуска. Конечно, это связано с целым рядом трудностей, но, не преодолев их, трудно рассчитывать на успех.

В целом же хочется отметить положительную роль конкурса. Да, именно так! Несмотря на то что учебника у нас по-прежнему нет. Важно, что были привлечены к активной деятельности те люди, которые потенциально могут создать добротный учебник. Мы познакомились с целым рядом интересных авторских коллективов, и, хотя их первая попытка на новом поприще не увенчалась успехом, надеюсь, что к следующему конкурсу они подойдут во всеоружии.

108

Кормление по компьютеру

Специалисты Болгарии разрабатывают и внедряют в рамках СЭВ электронные системы для сельского хозяйства. Вот некоторые направления:

контроль за работой сеялок, включая учет нормы высева;

контроль за количеством внесенных гербицидов;

измерение обработанной площади;

измерение потерь при уборке урожая;

автоматическое вождение сельскохозяйственных машин по рядам убираемой культуры;

использование иерархической микропроцессорной системы для комплексной автоматизации работ на молочном комплексе.

Последняя система обеспечивает автоматический сбор информации о продуктивности и здоровье каждой коровы, а также кормление концентрированным кормом в увязке с индивидуальными удоями молока, группировку коров в связи с периодом лактации.

ЧТО МОЖЕТ ЭВМ?

Мультфильмы и компьютеры

Соавтором многих мультфильмов будущего станет ЭВМ. Самое трудное — «оживить картинки». В Канаде создан вычислительный мультипликационный комплекс, «оживляющий» трехмерные модели объектов. Основной принцип моделирования — представление поверхности в виде многоугольников. На экране дисплея эти многоугольники могут быть сглажены, и

тогда появляются плавные очертания. Мягкие объекты определяются как набор управляемых точек, каждая из которых представляется сферой. Когда точки располагаются рядом, получаются огонь, облака и другие мягкие поверхности. Художник создает такую модель, чтобы ее можно было рассматривать в увеличенном виде. Данные о ней накапливаются в памяти ЭВМ, с информацией о перспективе и расположении на экране. Сложные тела запоминаются в виде составных частей (руки, ноги, торс человека). Движение и освещенность определяются для каждой части модели. Сложное движение разлагается на перемещение, вращение и масштабирование, которые выполняются соответствующей программой. Объект может вращаться вокруг собственной оси, сжиматься и расширяться, плавно исчезать и появляться, становиться прозрачным.

Так что через несколько лет художники-мультипликаторы, не владеющие информационной культурой, рискуют остаться без работы.

А. ДУВАНОВ, Я. ЗАЙДЕЛЬМАН, Ю. ПЕРВИН

Роботландия

В летней региональной школе юных программистов (Переславль-Залесский, август, 1987) среди прочих были организованы учебные группы, в которых проходили занятия с младшими школьниками по специальной программе. Основной задачей преподавателей являлась экспериментальная проверка программно-методического комплекса «Роботландия»*, развивающего подход к раннему обучению информатике, предложенный группой новосибирских исследователей [1, 2].

Занятия на базе программно-методического комплекса «Роботландия», по мнению авторов, должны предшествовать систематическому изучению инфор-

матики (например, на базе языка Лого). В начальной стадии обучения «Роботландия» привлекательна тем, что, в отличие от Лого, сразу задает ребенку видимую среду обитания исполнителя. 109

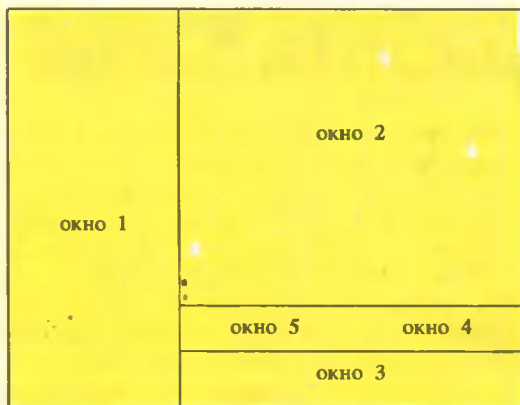
В «Роботландию» входит набор программно реализованных исполнителей-роботов. Каждый исполнитель имеет свою систему команд, рассчитанную на действия робота в специфичной для него среде обитания. Управлять исполнителем можно либо непосредственно путем выдачи команд из его системы предписаний, либо с помощью программ, составленных на языке Роботландия. Формальное описание языка приводится ниже.

```
⟨программа⟩ ::= ⟨процедура⟩ | ⟨программа⟩ ⟨процедура⟩
⟨процедура⟩ ::= {ПРОЦ | ПРОЦЕДУРА} ⟨имя⟩ ⟨тело⟩ {КОН | КОНЕЦ}
⟨тело⟩ ::= ⟨структура⟩ | ⟨тело⟩ ⟨структура⟩
⟨структура⟩ ::= ⟨вызов процедуры⟩ | ⟨развилка⟩ | ⟨цикл⟩
⟨вызов процедуры⟩ ::= ⟨имя⟩ [⟨список параметров⟩]
⟨развилка⟩ ::= ЕСЛИ ⟨условие⟩
                ТО ⟨вызов процедуры⟩
                ИНАЧЕ ⟨вызов процедуры⟩
⟨цикл⟩ ::= ЦИКЛ ⟨целое⟩ ⟨вызов процедуры⟩
```

Язык Роботландия отличается минимальным числом базовых структур, простотой и естественностью грамматики, что не мешает составлять доста-

точно сложные, хорошо структурированные программы. Как заметил один из иностранных гостей летней школы, он похож на гавайский язык, в алфавите которого всего 11 букв. Тем не менее гавайцы умудряются строить весьма сложные языковые конструкции.

* Программно-методический комплекс «Роботландия» разработан в лаборатории школьной информатики ИПС АН СССР.



110

В «Роботландии» несколько непохожих друг на друга роботов, но интерфейс между ребенком и каждым исполнителем является единым в системе. Это позволяет экономить время при переходе от одного исполнителя к другому. Экран компьютера устроен одинаково, независимо от того, с каким исполнителем работает ребенок (рис. 1).

Окно 1 — поле программы — служит для набора, редактирования программы; с него текст выводится на печать. Поле имеет размеры 150×20 . Окно показывает часть поля размером 24×20 .

Окно 2 — среда обитания исполнителя — служит для начальной установки исполнителя и наблюдения за ходом выполнения отдельных команд и программ.

Окно 3 — поле исполнения команд и программ. В нем можно использовать команду или процедуру в двух режимах: основном и отладочном (по шагам).

Окно 4 — восстановление начальной установки после работы исполнителя.

Окно 5 служит для записи, чтения и присоединения программ из библиотеки исполнителя на диске.

«Роботландия» в учебной деятельности летней школы

Ко времени экспериментальной проверки «Роботландии» в летней школе полностью были завершены два исполнителя: Таракан и Корректор [3, 4]. Остальные — Раскрашка, Шарманщик, Муравей, Машинист — были выполнены в виде макетных образцов и работали лишь в режиме непосредственного

(не программного) управления. На занятиях в летней школе предусматривалась проверка этих программных продуктов как на обычную живучесть по отношению к некорректным действиям пользователя, так и на удобство в работе, методическую ценность.

После занятий, учитывая опыт прошедшего дня, разработчики модифицировали свои программы. Иногда изменения вносились даже в систему команд робота. Так, команда ПИСАТЬ α из системы команд Корректора была заменена командой ПИШИ α ; строчные русские и латинские буквы алфавита Корректора были заменены на прописные, что уменьшило число переключений регистра; квадрат курсора размером 8×8 был увеличен до размера 12×12 и т. п.

Авторы стремились, чтобы помимо усвоения основ информатики и программирования, приобретения навыков работы с ЭВМ как с удобным инструментом при создании, хранении, поиске и обработке информации (электронные таблицы, редакторы, банки данных и т. д.), «Роботландия» способствовала проявлению творческой активности ребенка, закреплению знаний, навыков и умений, получаемых при изучении школьных дисциплин. Это достигается за счет разнообразия исполнителей (Художник, Музыкант, Математик и др.), наличия в комплексе других программных средств, не являющихся роботами-исполнителями (обучающе-моделирующие программы: Правила поведения, Прохождение маршрута по плану и др.).

В программно-методическом комплексе предусмотрены компьютерные конференции и коллективный выпуск компьютерной газеты...

Разумеется, весь программно-методический комплекс, рассчитанный на работу в общеобразовательной школе с III по X класс, не мог быть полностью проверен в условиях летней школы. С сентября 1987/88 учебного года несколько переславских школ включились в педагогический эксперимент по раннему обучению информатике. В рамках этого эксперимента «Роботландия» уделено важное место. Поэтому предшествующее учебному году опробование комплекса

является важным этапом подготовки его к внедрению. Переславский эксперимент предусматривает разные начальные возрастные стартовые позиции (III и V классы). Половина экспериментальных классов начинает изучение информатики с исполнителей «Роботландии», другая половина — непосредственно с языка Лого. Различна подготовка преподавателей, участвующих в эксперименте, — учителя информатики общеобразовательных школ и сотрудники лаборатории школьной информатики Института программных систем, разработавшие программный и методический инструментальный «Роботландии». Экспериментальные классы работают в различных кабинетах информатики, оснащенных отечественными и зарубежными персональными компьютерами.

Организация учебных занятий

В группу, работающую с «Роботландией» в летней школе, входили дети 7—12 лет. Занятия вел старший преподаватель А. А. Дуванов с ассистентами Ю. А. Первиным, Я. Н. Зайдельманом и Ю. Ю. Рупасовым.

Вначале занятия проходили ежедневно по 3 часа (с 10.00 до 13.00), но к концу первой недели выяснилось, что этот режим неудобен. Детей было много (12 — в первые дни занятий, 19 — в последние), а компьютеров мало (семь ПЭВМ «Ямаха»). В результате — теснота в зале для занятий. Дети мешали друг другу, трудно было проводить практические занятия. Оказалось также, что 3 часа даже интересных занятий с компьютером для детей этого возраста слишком утомительны (что было подтверждено группой сотрудников НИИ физиологии детей и подростков АПН СССР, проводивших систематические наблюдения за группой).

В последующие дни режим учебной работы изменили. Дети были разделены на две подгруппы. Первая занималась с 10.00 до 11.30, вторая — с 11.30 до 13.00. Таким образом, обязательное учебное время сократилось вдвое. Этот «дефицит» частично компенсировался во второй половине дня, когда проводились консультации, дети (по желанию) могли

порешать задачи, составить свои программы. Однако из-за вынужденного сокращения ежедневных занятий и полной отмены их в дни общешкольных мероприятий первоначально намеченный план полностью выполнен не был. Тем не менее мы считаем, что в результате проведенных занятий ученики получили достаточно полный и целостный набор начальных представлений об ЭВМ и информатике, а преподаватели — ценный материал для подготовки к дальнейшим экспериментам.

Первые встречи с исполнителями

Путешествуя по Роботландии — стране, населенной разнообразными роботами-исполнителями, дети осваивают понятие «информация», многообразие форм ее представления, узнают об областях применения компьютеров, изучают основы программирования.

С исполнителями дети знакомятся на примерах из непосредственного окружения.

Учитель говорит: «Исполнителем мы назовем устройство, которое нас слушается. Вот телевизор. Меня он слушается, а младшего брата — нет. Чтобы научиться управлять исполнителем, нужно выучить его систему команд и точно знать, что он сделает в результате команды».

Первым исполнителем, с которым встретились ученики, был Мудрый Кролик. Эту игру-головоломку мы планируем включить в «Роботландию» в качестве конструктора игр. Она заимствована из штатного программного обеспечения Микрософт MSX. Кролик живет в лабиринте на экране компьютера, умеет передвигаться по лабиринту и переносить разбросанные мешки, соблюдая ряд простых правил. Цель игры — перенести все мешки в специально отведенные места. Управляется Кролик в непосредственном режиме нажатием клавиш, перемещающих курсор. Работая с Кроликом, дети осваивают понятие формального исполнителя, точно и беспрекословно выполняющего любую правильную команду.

Важно, чтобы дети усвоили, что понятия о правильной команде у исполнителя и управляющего им человека не совпада-

ют. Кролик «считает» команду правильной, если она входит в его систему предписаний и выполнима в конкретной ситуации. К этому и сводится вся его «мудрость». Поэтому Кролик с одинаковой готовностью перетаскивает мешки на место и загоняет их в тупик. Для человека же два этих действия не равнозначны.

Одно из методических достоинств Кролика заключается в следующем: для успешного решения задачи необходимо тщательно спланировать свои действия и довести их до уровня примитивных команд, понятных исполнителю.

Преподаватели отмечали суть исполнителей как универсальных обработчиков информации. Для Мудрого Кролика исходной информацией является конфигурация лабиринта, начальное положение мешков и точек, в которые их нужно передвинуть, а также начальное положение самого Кролика. Обработка информации заключается в изменении положения мешков и Кролика. Цель обработки — результирующая информация — наличие мешков в запланированных местах. Кроме 60 стандартных головоломок-лабиринтов Кролик позволяет создавать собственные варианты заданий. Здесь дети впервые встречаются с редактированием информации.

После изучения Кролика были проведены два конкурса: на лучшее решение одной из стандартных головоломок и на создание нового задания-лабиринта. С предложенной головоломкой справились все ученики. Однако количество потребовавшихся для решения ходов менялось в довольно широких пределах. Интересно, что корреляция между возрастом учеников и скоростью решения оказалась очень слабой. Большой интерес вызвал у детей второй конкурс. Многие сконструировали по 3—4 лабиринта. Среди придуманных головоломок встречались довольно интересные и содержательные.

Следующий исполнитель, с которым познакомились дети, — Планетоход — серийно выпускаемая электронно-механическая игрушка с программным управлением. Здесь существовало, что исполнитель представлен не на экране, а в материальном мире. Так мы стреми-

лись расширить представление об исполнителях. Важнейшее методическое достоинство Планетохода заключается в том, что составление программы предшествует исполнению, а также в простоте программирования и наглядности получаемых результатов. Необходимо не только тщательно спланировать действия исполнителя, как это было при работе с Кроликом, но и ввести полную программу до начала ее выполнения. Корректируя программу, добиваясь точного прохождения Планетоходом заданной трассы, дети получают первые представления о процессе отладки.

Занятия с Планетоходами вызвали большой интерес у детей. Они увлеченно придумывали задачи для себя и своих товарищей, часто просили разрешения поиграть в свободное время.

Наигравшись с Планетоходом, дети вернулись к экранам компьютеров. Новые исполнители — Раскрашка и Шарманщик — помогли продолжить изучение процессов создания и редактирования информации, начатое при работе с Кроликом, расширить представления детей о возможностях ЭВМ.

Дети пишут музыку, рисуют, выпускают газету

Программа Раскрашка, разработанная в лаборатории школьной информатики ИПС АН СССР Ю. Ю. Рупасовым, — простейший графический редактор. По экрану перемещается шаблон. Нажатием специальных клавиш можно изменять его форму и размеры, оставлять след шаблона и закрашивать его произвольным цветом. Работая с Раскрашкой, дети знакомятся с графической формой представления информации. После изучения Раскрашки был проведен конкурс на лучший компьютерный рисунок. Дети приняли активное участие в конкурсе, многие сдавали по 3—4 рисунка. Отпечатанные на принтере рисунки были выставлены в учебном классе.

Программа Шарманщик, разработанная в лаборатории школьной информатики ИПС АН СССР Н. Б. Дроздовым, А. А. Дувановым, — простейший музыкальный робот. Предварительно детей познакомили с нотным письмом, его

историей. На занятиях использовался некомпьютерный исполнитель — детский электронный музыкальный инструмент «Соловушка». Шарманщика нельзя рассматривать в качестве программы обучения сольфеджио. Шарманщик живет в Роботландии, и главная его задача — обучение информатике. Он знакомит детей с новой формой представления информации. В частности, нотная запись простейших мелодий представляется как программа-задание исполнителю Шарманщику. «Музыкальная» часть этого робота-исполнителя пока предельно упрощена: отсутствие знаков альтерации заставляет работать только в тональности до мажор.

В отличие от традиционных редакторов музыкальный робот предполагает не только создание информации, но и ее исполнение. Таким образом, работая с Шарманщиком, дети делают очередной шаг, приближающий их к пониманию основ программирования. С этой точки зрения интересна роль Обезьянки, которая бежит по нотному стану, вставляет и стирает ноты, корректирует их высоту и длительность. Сам же Шарманщик только исполняет мелодию. Здесь предельно наглядно выражено различие между текстом программы (программой является нотная запись мелодии) и ее исполнением.

Познакомившись с Шарманщиком, дети набрали, отредактировали и исполнили (вместе и по отдельности) несколько простейших хорошо знакомых мелодий — «Чижик», «Маленькой елочке холодно зимой», «Где ты, колечко?», «Веселые гуси», а потом организовали конкурс музыкальных произведений. Следует отметить, что большинство детей впервые познакомились с нотной грамотой благодаря Шарманщику.

Текстового исполнителя в Роботландии пока нет. Работа с текстом для младших детей не очень привлекательна. Поэтому работа с текстовым редактором проходила факультативно, во внеучебное время. Была выпущена компьютерная газета «Роботландские новости», полностью подготовленная детьми. Все заметки набирали с помощью редактора «Микромир», а иллюстрации были подготовлены в Раскрашке.

Следующие два исполнителя — Муравей и Машинист — подвели итог первому этапу работы. Идея этих исполнителей заимствована из работ Г. А. Звенигородского [1], программная реализация для ПЭВМ «Ямаха» подготовлена в лаборатории школьной информатики ИПС АН СССР (авторы И. Г. Лилитко, В. Ф. Забавина).

Исполнитель Машинист моделирует классическую железнодорожную ситуацию. Имеются пути со стрелками, вагоны и локомотив. Требуется переставить вагоны и получить заданную расстановку. Работа ведется в непосредственном режиме, команды задаются набором слов с клавиатуры и немедленно исполняются. Машинист закрепляет полученные ранее знания об исполнителях, информации и программах, знакомит с новыми формами представления информации, демонстрирует применение компьютера как инструмента моделирования и анализа ситуаций.

Исполнитель Муравей живет на клетчатом поле, по которому разбросаны кубики с буквами. Муравей передвигается по полю, толкая кубики. Задача — собрать из кубиков разбросанное слово. Легко заметить, что Муравей — близкий родственник Кролика. Правда, управлять Муравьем намного легче, потому что он живет в открытом поле, а не в сложном лабиринте, но зато у Кролика все мешки одинаковы, а для Муравья порядок кубиков не безразличен. Муравей управляется в непосредственном режиме, но в процессе работы формируется протокол в виде листинга, содержащий все выполненные команды.

В центре Роботландии живет Таракан

Муравей подготовил детей к работе с более сложным исполнителем — Тараканом. Среда обитания и система предписаний Таракана и Муравья практически полностью совпадают, отличия — в методах управления этими исполнителями. В опробованной версии системы Муравей управлялся непосредственно, без использования механизма процедур. Для него возможны толь-

ко линейные программы. Таракан понимает язык Роботландия в полном объеме, программа состоит из множества взаимодействующих процедур, каждая из которых может иметь сложную внутреннюю структуру. Один из новых объектов Таракана — перевернутые кубики, которые изображаются в среде обитания исполнителя вопросительным знаком. Таракан может увидеть букву на перевернутом кубике, только толкнув его. Одна из управляющих клавиш (SELECT) используется для перемешивания перевернутых кубиков.

Изучение Таракана проводилось в следующей последовательности:

1. Система предписаний и среда обитания исполнителя. Управление Тараканом в непосредственном режиме.

114 2. Программное управление. Понятие процедуры. Вложенные вызовы процедур.

3. Понятие цикла. Циклические процедуры Таракана.

4. Понятие ветвления. Условия, проверяемые Тараканом. Сложные ветвления.

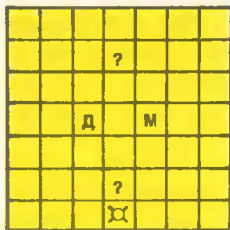
5. Понятие рекурсии. Рекурсивные процедуры Таракана. Бесконечная и конечная рекурсия.

Понятие процедуры вводилось как описание команды, не входящей в систему команд исполнителя. Рассмотрим этот подход на примере задачи построения дома (рис. 2).

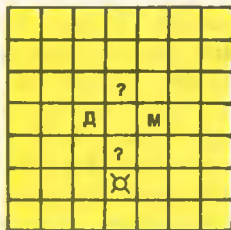
Задача. Помогите Таракану построить дом. (Буква «О» находится на одном из перевернутых кубиков.)

Решение. Общий план: толкаем вверх ближайший кубик. Если на кубике — буква О, делаем еще один толчок вверх, и дом построен. Если на кубике посторонний символ, то О находится на дальнем кубике и нужно обойти дом сбоку.

2



3



ЭТО ДОМ
ВВЕРХ
ЕСЛИ О
ТО ВВЕРХ
ИНАЧЕ ???
КОНЕЦ

На одной ветви условного предписания команда Таракана ВВЕРХ. На другой должна стоять более сложная команда, заставляющая исполнителя обойти дом сбоку и толкнуть вниз нужный кирпич. Такой команды нет в системе команд Таракана. Мы назовем ее ПОХОД и объясним Таракану при помощи описания ЭТО ПОХОД ... КОНЕЦ. Структура составной условной команды, на ветвях которой можно записывать только по одной команде, побуждает школьников к «запроцедурированию» своих действий, делает использование процедур необходимым и естественным.

ЭТО ДОМ
ВВЕРХ
ЕСЛИ О
ТО ВВЕРХ
ИНАЧЕ ПОХОД
КОНЕЦ
ЭТО ПОХОД
ЦИКЛ 2 ВЛЕВО
ЦИКЛ 4 ВВЕРХ
ЦИКЛ 2 ВПРАВО
ЦИКЛ 2 ВНИЗ
КОНЕЦ

Условная команда оказалась необходимой, поскольку при составлении программы заранее не было ясно, на каком из кубиков находится буква О, и, следовательно, надо предусмотреть обе возможности. Первые упражнения с условной командой проводятся на задачах с одним перевернутым кубиком. Учитель (или ассистент) просит ученика отвернуться от экрана и устанавливает перевернутый кубик. В другой раз кубики переворачивают ученики, по-прежнему попросив товарища отвернуться. Далее в задачах с двумя и более перевернутыми кубиками (такой задачей является, в частности, и задача на строительство тараканьего дома) каждый ученик осуществляет расстановку кубиков сам. Случайный характер расстановки обеспечивается перемешиванием (нажатием клавиши SELECT). Впоследствии, занимаясь отладкой напи-

санных ими программ, школьники убеждаются в необходимости проверки всех ветвей, возникающих при использовании условных команд в программе.

В задаче о строительстве дома использована так называемая полная форма условной команды — с обеими ветвями. Сокращенная форма условной команды возникает в следующей задаче, которая преподносится школьникам как «упрощение» предыдущей: надо построить такой же дом, но его кирпичи (один из которых содержит требуемую для строительства букву O) расположены ближе к месту постройки (рис. 3).

В этой задаче на ветви ТО не надо выполнять никаких действий:

```

ЭТО ДОМ
ВВЕРХ
ЕСЛИ О
ТО
ИНАЧЕ ОБХОД
КОНЕЦ
ЭТО ОБХОД
ЦИКЛ 2 ВПРАВО
ЦИКЛ 3 ВВЕРХ
ЦИКЛ 2 ВЛЕВО
ВНИЗ
КОНЕЦ

```

Наконец, вводится второй вид сокращенной условной команды — с отсутствующей ветвью ИНАЧЕ. В этом случае не указывается даже слово ИНАЧЕ. Так, если в предыдущей задаче на перевернутых кубиках поставлены буквы Ы и O, то решение можно записать:

```

ЭТО ДОМ
ВВЕРХ
ЕСЛИ Ы
ТО ОБХОД
КОНЕЦ

```

Корректор, рекурсии

Корректор — исполнитель, который, подобно машине Тьюринга, работает с длинной лентой, разбитой на ячейки. Кроме ленты Корректору доступна еще одна ячейка памяти, условно называемая ящиком. Система команд работа имеет следующий вид:

```

ПИШИ α — записать в обозреваемую
ячейку на ленте символ α;
ВПРАВО — шаг исполнителя вдоль
ленты вправо;
ВЛЕВО — шаг исполнителя вдоль
ленты влево;

```

ЯЩИК+ — копировать символ с ленты в ящик;

ЯЩИК— — копировать символ из ящика на ленту;

ПЛЮС — заменить символ на ленте на следующий в смысле алфавита Корректора;

МИНУС — заменить символ на ленте предыдущим в смысле алфавита Корректора;

ГОЛОС — подача звукового сигнала. Корректор располагает условной командой, которая позволяет сравнивать взаимное положение в алфавите двух символов — символа в текущей ячейке ленты с символом в ящике.

Алфавит Корректора:

```

0 1 2 3 4 5 6 7 8 9 A B B Г Д Е Ж З
И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш
Щ Ъ Ы Ь Э Ю Я A B C D E F
G H I J K L M N O P Q R S T U
V W X Y Z — + / * () [] . ,
! ? ; : ' " % &

```

Символ «^» — обозначение пустого поля на ленте.

Корректор — сложный исполнитель, позволяющий ставить и решать весьма содержательные задачи. К сожалению, ограниченное время работы летней школы не позволило провести полный курс изучения Корректора. Было проведено лишь 3 занятия: система команд и среда обитания Корректора; работа с процедурами (в том числе, рекурсивными) и ветвлениями; работа Корректора с ящиком.

Надо отметить, что управление Корректором было мало понятно, и, следовательно, не очень интересно младшим детям. Зато старшие, особенно самые сильные, которым наскучил Таракан, принимали Корректора с восторгом. Показателен тот факт, что задачи Корректора с увлечением решали старшие дети из других групп.

В качестве примера приведем одну серию задач Корректора.

Задача 1. На ленте находятся 2 символа: непосредственно перед Корректором и справа от него. Требуется поместить в ящик больший из них — старший по алфавиту.

Решение очевидно:

```

ЭТО выбор — из — двух
ЯЩИК+

```

ВПРАВО
ЕСЛИ <
ТО ЯЩИК+
КОНЕЦ

Задача 2. Сделать то же самое, но на ленте уже не 2, а 3 символа.

Решение:

ЭТО выбор — из — трех
ЯЩИК+
ВПРАВО
ЕСЛИ <
ТО ЯЩИК+
ВПРАВО
ЕСЛИ +
ТО ЯЩИК+
КОНЕЦ

Задача 3. Выбрать наибольший из 10 символов.

Решение:

ЭТО выбор — из — десяти
ЯЩИК+
ЦИКЛ 10 выбор
КОНЕЦ
ЭТО выбор
ВПРАВО
ЕСЛИ <
ТО ЯЩИК+
КОНЕЦ

Задача 4. Выбрать наибольший из про-

извольного числа непустых символов.

Решение:

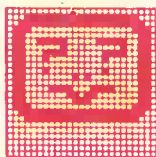
ЭТО большой — выбор
ЯЩИК+
ВПРАВО
вперед
КОНЕЦ
ЭТО вперед
ЕСЛИ <
ТО ЯЩИК+
ВПРАВО
ЕСЛИ
ТО ГОЛОС
ИНАЧЕ вперед
КОНЕЦ

Литература

1. Звенигородский Г. А. Первые уроки программирования. М., 1985.
2. Еришов А. П., Звенигородский Г. А., Первин Ю. А. Школьная информатика (концепции, состояние, перспективы). Новосибирск, 1979.— (Препринт ВЦ СО АН СССР).
3. Дуванов А. А., Немилостива В. В., Какаулин О. Г. Введение в программирование. Благовещенск, 1986.
4. Дуванов А. А. Информатика без ЭВМ// Информатика и образование. 1987. № 3.

116

Веселый
урок



Я. БОКАНС

Латвийский государственный университет

А. ГРИБУСТС

Ср. школа № 1 пос. Лиелварде Латв. ССР

Не только на уроке информатики

Мы уже сообщали о создании КВТ на базе персональных компьютеров «Электроника БК-0010» в средней школе № 1 поселка Лиелварде Латвийской ССР (Информатика и образование. 1986. № 2). В 1986/87 учебном году в нем проводились практические занятия по информатике, а также кружковая и индивидуальная работа с учащимися.

В течение 1986/87 учебного года созданы компьютерные программы и техническое оснащение для конкурсов, соревнований по баскетболу, а также автоматизированный комплекс для экспериментальной проверки законов механики. С программами может работать любой ученик или учитель, имеющий навыки ведения диалога с компьютером.

Одна из форм внеклассной работы — проведение тематических конкурсов. Для оперативной обработки их результатов мы применяем БК-0010 с демонстрационным монитором. Программа «Электронное табло» реализует следующие функции:

- запоминание названий участвующих в конкурсе команд и этапов конкурса;
- хранение результатов каждой команды в отдельных этапах и подсчет суммы очков;

- распределение мест команд по сумме очков;

- отображение суммы очков, результатов каждого этапа (или распределения мест) в форме таблицы на экране демонстрационного монитора.

До начала конкурса в память ЭВМ вводятся названия команд и этапов конкурса, в течение конкурса — результат каждой команды в данном этапе.

На рис. 1 показан экран монитора после этапа В (название этапа «Песня»). В конкурсе принимают участие шесть команд. В первой графе таблицы — названия команд, во второй — результат этапа В, в третьей — сумма очков.

КОМАНДА	В "ПЕСНЯ"	СУММА ОЧКОВ
1. 7А КЛАСС	5	12
2. 7Б КЛАСС	5	13
3. 7Ц КЛАСС	4	12
4. 6А КЛАСС	4	14
5. 6Б КЛАСС	4	11
6. 6Ц КЛАСС	3	11

КОМАНДА-■

117

На рис. 2 показан экран монитора после завершения конкурса. Во второй графе отображено распределение мест по сумме очков. Нижняя часть экрана отведена для ввода результата текущего этапа и директив управления программой. Предусмотрены следующие директивы:

- вызов любого этапа (этапы обозначаются латинскими буквами);

- запрос ввода результата для выбранной команды (команды обозначаются цифрами);

- вывод на экран монитора списка названий этапов;

- распределение мест по сумме очков.

Программа «Электронное табло» успешно прошла испытания в полит-

КОМАНДА	МЕСТО	СУММА ОЧКОВ
1. 7А КЛАСС	3	12
2. 7Б КЛАСС	2	13
3. 7Ц КЛАСС	3	12
4. 6А КЛАСС	1	14
5. 6Б КЛАСС	4	11
6. 6Ц КЛАСС	4	11

ЭТАП-■

2

конкурсе старшеклассников Огрского района Латвийской ССР и в конкурсах пионерской дружины школы.

Для обслуживания школьных соревнований по баскетболу служит программа «Баскетбольный протокол». В спортзале устанавливается компьютер БК-0010 и демонстрационный монитор. Программа реализует следующие функции:

- запоминание названий команд, фамилий и номеров игроков;
- подсчет результата игры;
- хранение информации о результативности и количества фолов (персональных замечаний);

- отображение на экране демонстрационного монитора текущего счета матча;
- вывод на экран монитора таблицы результативности игроков;

118 подача звуковой сигнализации при превышении лимита фолов.

До начала игры судья-секретарь вводит в память компьютера названия команд, фамилии и номера игроков. Эти данные можно подготовить заранее в виде файла данных и ввести в память компьютера с магнитной ленты.

Для управления программой «Баскетбольный протокол» предусмотрены следующие директивы:

- запрос ввода номера игрока и количества полученных очков при забитом мяче;

- запрос ввода номера игрока команды при получении фола;

- вывод на экран монитора таблицы статистики;

- замена номера периода матча.

Программа «Баскетбольный протокол»

применялась в соревнованиях по баскетболу школьной спартакиады.

Существенную роль в ознакомлении учащихся с применением ВТ на производстве играют учебные стенды автоматизированного управления, сбора и обработки данных на базе ЭВМ. Важно дать ученику возможность самостоятельно поработать на таком стенде, почувствовать преимущества систем на базе ЭВМ перед ручным управлением, регистрацией и обработкой данных. Подходящей областью является автоматизация работ физического практикума средней школы. На базе компьютера БК-0010 создан автоматизированный комплекс для экспериментальной проверки законов механики. Комплекс состоит из прибора демонстрации законов механики (ПДЗМ), персонального компьютера БК-0010, блока сопряжения и блока питания. Общий вид комплекса показан на рис. 3, блок-схема — на рис. 4.

Используя программированный интерфейс компьютера БК-0010, осуществляется управление движением платформы ПДЗМ. Блок сопряжения предназначен для питания обмоток электромагнитов.

Компьютер, выполняя программу управления экспериментом, реализует следующие функции:

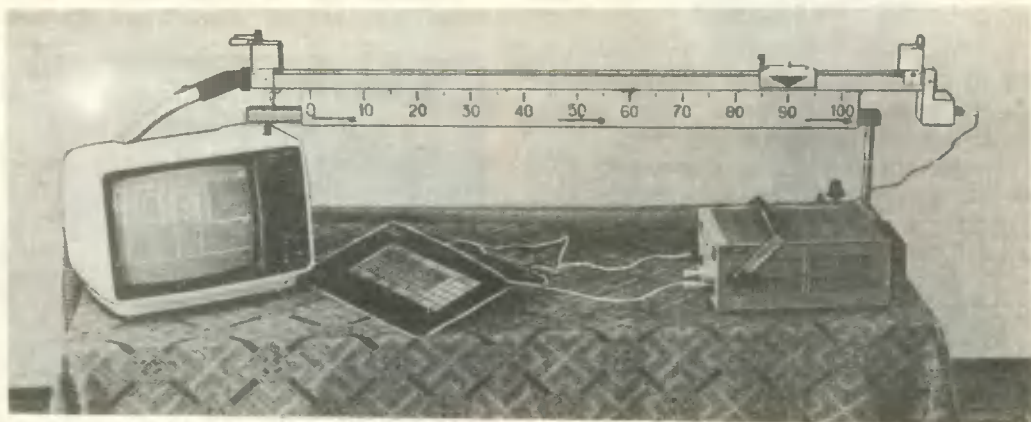
- измерение интервалов времени;

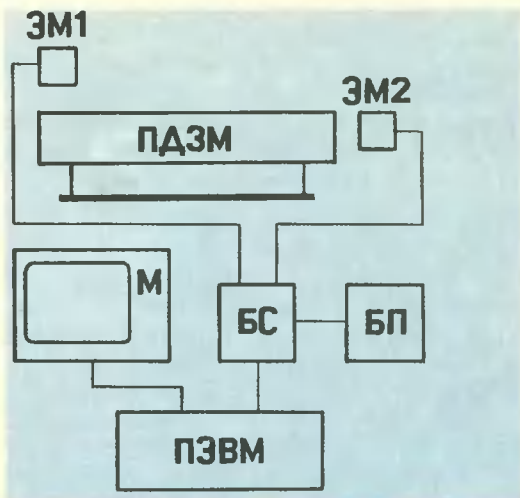
- предварительная обработка данных;

- отображение результатов эксперимента на экране монитора в форме таблицы и графика.

С помощью комплекса можно проводить следующие эксперименты:

3





4

исследования зависимости перемещения от времени в равноускоренном движении;

исследование зависимости конечной скорости от времени в равноускоренном движении;

проверка второго закона Ньютона.

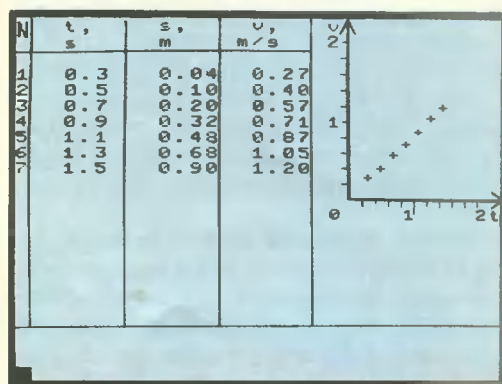
Рассмотрим порядок проведения эксперимента «Исследование зависимости конечной скорости от времени в равноускоренном движении». После подготовки комплекса к работе задается количество измерений. Каждое измерение начинается с задания времени движения платформы. Компьютер выключает электромагнит ЭМ1 (отпускает платформу) и после указанного интервала времени включает электромагнит ЭМ2 (останавливает платформу). После этого экспериментатор считывает с линейки пройденное платформой расстояние и вводит в компьютер. Программа вычисляет конечную скорость платформы и на экране монитора отображает результат в таблице и на графике (см. рис. 5).

Автоматизированный комплекс для экспериментальной проверки законов

механики использовался на уроках физики в IX классе.

Разработка проектов программ проведена совместно с кафедрой физики полупроводников физико-математического факультета Латвийского государственного университета им. П. Стучки. Реализация программ и технической части автоматизированного комплекса проведена учениками старших классов (Я. Кловиньш, А. Лаздиньш, И. Грантс). Работы были представлены на 3-й Прибалтийской конференции Научного общества школьников и получили положительный отзыв.

Интерес старшеклассников и учеников младших классов к информатике вырос.



119

5

Ученики воспринимают компьютер не только как объект изучения в курсе «Основы информатики и ВТ», а как инструмент с широкими возможностями. Активизировалась кружковая и индивидуальная работа учеников.

Все программы составлены на Фокале, однако проекты будущих разработок предусматривают использование ассемблера. Надеемся, что описанные в статье разработки помогут учителям в организации кружковой работы и в создании подобных программ и комплексов для своих школ.

О первом всесоюзном турнире по компьютерной игре «Бой в памяти»

120

Игра «Бой в памяти» (Core War), предложенная профессором А. Дьюдни (США) [1—3], относится к чрезвычайно интересному классу компьютерных игр, требующих от играющего умения программировать. Играющие пишут программы, которые начинают жить собственной жизнью в таинственных лабиринтах машинной памяти. Авторам программ остается лишь болеть за своих питомцев. Эта особенность обуславливает разнообразие применений «Боя в памяти»: он может с успехом применяться для начального обучения программированию, для наглядной демонстрации приемов программирования, а также для интересного и полезного проведения досуга.

Краткое описание игры. Архитектура виртуальной машины. Игра проходит на некоторой виртуальной машине, имеющей 8000 ячеек оперативной памяти, замкнутых в кольцо. Ячейки нумеруются с 0 до 7999. После ячейки 7999 снова следует нулевая. Например, числа 1988 и 9988 задают один и тот же адрес памяти.

Монитор Марс. Он управляет игрой. Перед началом вся память виртуальной машины заполнена нулями. Монитор размещает две соревнующиеся программы в случайные участки памяти (не ближе 1000 ячеек одна от другой) и выполняет по очереди по одной команде из каждой программы. Программа представляет собой последовательность команд. Каждая команда занимает одну ячейку памяти. Как только монитор пытается выполнить нечто, не являющееся командой, он объявляет программу, в которой это произошло, проигравшей. Таким образом, цель каждой программы — внести в тело программы противника некорректный код (например, вместо команды записать число 0) и при этом защитить от повреждений себя.

Язык REDCOD. Система команд виртуальной машины (язык REDCOD)

представлена в таблице. Буквы X и Y обозначают наличие операндов в команде. Например, у команды ADD два операнда, а у команды JMP — один. При записи команд в память операнд X всегда пишется на место первого операнда, а операнд Y на место второго.

Команда DAT — неисполняемая. Более того, попытка выполнить ее ведет к катастрофическим последствиям для программы. Действие ее состоит в том, что перед выполнением программы в том месте, где она находится, будет записано число — ее операнд.

Команда JMG равносильна команде «переход, если не 0», так как отрицательных чисел в нашей машине не бывает.

Команда SPL предназначена для порождения новой программы. После ее выполнения программа продолжает работать, а по адресу ее операнда порождается новая программа, которая действует независимо от программы-родителя. Конечно, если по адресу, заданному в команде SPL, нет правильных команд, то порожденная программа сразу погиб-

DAT	Y	зарезервировать данные
MOV	X Y	переслать содержимое ячейки X в ячейку Y.
ADD	X Y	сложить содержимое ячеек X и Y и поместить результат в ячейку Y
SUB	X Y	вычесть содержимое ячейки X из содержимого ячейки Y и поместить результат в ячейку Y
JMP	X	передать управление на ячейку X
JMZ	X Y	передать управление на ячейку X, если в ячейке Y содержится нуль
JMG	X Y	передать управление на ячейку X, если в ячейке Y содержится число большее нуля
DJN	X Y	вычесть 1 из содержимого ячейки Y, и если при этом получится не нуль, то передать управление на ячейку X
CMP	X Y	сравнить содержимое ячеек X и Y, и если не равно, то пропустить следующую команду
SPL	X	породить новый процесс (новую программу)

нет. Забота об этом лежит на программисте. Очередность выполнения команд породившей и порожденной программ следующая. Допустим, что у играющего А работает одна программа (a), а у играющего В две программы (b1 и b2), причем программа b2 порождена программой b1 с помощью команды SPL. Очередность ходов в этом случае такова (считая от команды SPL): b1, a, b2, a, b1, a, b2, a...

Особая прелесть команды SPL в том, что в случае гибели одной из действующих программ данного игрока другие продолжают сражаться. Независимые программы и уничтожать надо независимо. Правда, за это приходится платить быстройдействием.

Остальные команды в комментариях не нуждаются.

Выполнение программы всегда начинается с первой исполняемой команды, т. е. в начале программы допускается одна или несколько команд DAT. Впрочем, это правило имеет исключение (о нем будет сказано ниже).

*Операнды и типы адресации.** Операндом команды может быть число без знака или со знаком «минус» с предшествующим признаком типа адресации. Существуют несколько типов адресации.

Непосредственная адресация (обозначается символом #) означает, что в качестве операнда используется число, записанное непосредственно после #.

Прямая адресация (признака не имеет) предписывает обратиться к ячейке, отстоящей на указанное число позиций от данной. Например,

MOV # 5 -1

пересылает число 5 в ячейку, находящуюся непосредственно перед данной командой. Команда

MOV 0 1

является самостоятельной программой. Она при каждом ходе копирует себя на ячейку вперед и заполняет собой всю память.

Косвенная адресация (символ @) предписывает к адресу команды прибавить число, стоящее после знака @

(полученный адрес обозначим через А), взять содержимое ячейки с адресом А (обозначим его N), вычислить значение $A+N$; это и есть искомый адрес.

DAT 0 (*)

DAT 25

DAT -2

MOV -2 @-1

Здесь команда MOV пересылает число 25 в ячейку, помеченную (*); адресация с автоинкрементом (автодекрементом). Знаки > (<) определяют еще один тип адресации. К адресу команды прибавляется число-операнд (обозначим результат А), содержимое ячейки А увеличивается (>) или уменьшается (<) на 1, новое содержимое ячейки А прибавляется к адресу команды. В результате получается искомый адрес. Знаки > и < могут стоять как до, так и после операнда. В первом случае увеличение (уменьшение) произойдет до вычисления адреса, как было описано, а во втором — после. Например:

DAT 0 (*)

DAT -1 (**)

MOV # 5 <-1

Здесь число в ячейке, помеченной (**), перед выполнением команды MOV уменьшится на 1 и станет равно -2, а число 5 будет переслано на -2 ячейки от команды MOV, т. е. в ячейку, помеченную (*).

Пример: программа, копирующая себя на сто ячеек вперед и передающая туда управление.

MOV # 7 -2 Счетчик команд для пересылки

MOV # 104 -2 Адрес пересылки

MOV @ -4 <-3 Пересылка команды и уменьшение адреса

DJN -1 -5 Уменьшение счетчика и переход на пересылку

ADD # 4 -5 Все переслано, установка адреса

JMP @-6 Переход

Операнд с непосредственной адресацией недопустим в качестве адреса перехода и второго операнда команд MOV, ADD, SUB и т. д., т. е. вся адресация в программах относительная. Программа не может работать с абсолютными адресами памяти.

Использование символических адресов. Пока мы записывали операнды толь-

* Названия типов адресации в языке REDCOD нестандартны, но автор сохранил терминологию А. Дьюдни.

ко числами. Это очень неудобно. Во-первых, их надо считать, а во-вторых, при изменении программы пересчитывать заново. Можно перед командами ставить символические имена-метки (метка должна состоять не более чем из восьми букв или цифр), тогда вместо адресов этих команд можно записывать их имена. Например, приведенную программу можно записать так:

```
COUNTER DAT 0
ADDRESS  DAT 0
          MOV #7      COUNTER
          MOV #104    ADDRESS
LOOP     MOV @COUNTER <ADDRESS
          DJN LOOP    COUNTER
          ADD #4      ADDRESS
          JMP @ADDRESS
```

Метка, код операции и операнды отделяются друг от друга хотя бы одним пробелом. Есть одна особая метка — START. Если она присутствует в программе, то выполнение программы всегда будет начинаться с команды, помеченной этой меткой.

Представление данных в памяти. Все числа в памяти виртуальной машины представлены по модулю 8000. Например, число —1 будет представлено как 7999.

Команды представляются 11-разрядными десятичными целыми числами:

код операции	1 десятичный разряд,
признак адресации операнда А	1 десятичный разряд,
признак адресации операнда В	1 десятичный разряд
операнд А	4 десятичных разряда,
операнд В	4 десятичных разряда.

Коды операций соответствуют расположению команд в таблице (у команды DAT код 0, у команды MOV — 1 и т. д.).

Признаки адресации кодируются так:

- # — 0
- 1
- @ — 2
- > А — 3
- А > — 4
- < А — 5
- А < — 6

Правила игры. Поединок продолжается до тех пор, пока у обеих сторон есть хотя бы по одной действующей программе. Как только погибнет последняя программа одной из сторон, она объявляется проигравшей. Правилами турнира устанавливается некоторое пороговое число выполняемых каждой программой команд.

Если программы выполняют условленное количество команд, но победитель не выявится, то фиксируется ничья. В нашем турнире пороговое число равно 15 000.

В остальном правила предельно просты. Допускается делать все, что позволяет язык REDCOD. Многочисленные примеры программ можно найти в [1—3].

Первый всесоюзный турнир. Институт программных систем АН СССР планирует провести всесоюзный турнир по «Бюю в памяти». В программе турнира — соревнование программ и семинар по компьютерным играм в обучении.

Желающие принять участие в турнире и (или) семинаре должны выслать в адрес оргкомитета тексты программ и (или) тезисы доклада. Количество программ для турнира от одного участника не ограничивается. В турнире могут принять участие все желающие. Никаких профессиональных, возрастных и других ограничений не вводится. Победителей ждут призы.

Программы для участия в турнире и тезисы докладов следует выслать до пятнадцатого мая 1988 г. по адресу: 152140, г. Переславль-Залесский, Институт программных систем АН СССР, оргкомитет первого всесоюзного турнира по «Бюю в памяти».

Литература

1. Дьюдни А. К. В игре «Бюю в памяти» компьютерные программы ведут битву за биты // В мире науки. 1984. № 7.
2. Дьюдни А. К. Обзор читательских откликов на статьи, посвященные игре «Бюю в памяти» // В мире науки. 1985. № 5.
3. Дьюдни А. К. Анализ результатов первого турнира по компьютерной игре «Бюю в памяти» // В мире науки. 1987. № 3.

А. МАСЛОВ

О содержании курса информатики

В настоящее время нет общепринятого определения информатики. Рассмотрим несколько определений информатики в порядке возрастания общности.

В пробном учебном пособии [1] сказано: «Информатика исследует законы и методы переработки и накопления информации... Ее развитие связано с появлением электронно-вычислительных машин». В программе курса «Основы информатики и вычислительной техники» для объявления конкурса на создание учебника для учащихся общеобразовательной школы это определение уточняется: «Информатика определяется как научная дисциплина, имеющая своим предметом изучение законов и методов накопления и обработки информации с помощью ЭВМ». В учебном пособии для преподавателей СПТУ [2] явного определения информатике не дается; основное внимание уделяется изучению методов описания производственных и социальных процессов. Методы описания природных процессов также иногда относят к информатике (например, говорят о генной информации, об обмене сообщениями между муравьями.) В рукописи конкурсного учебника [3] информатика определяется как «наука об информационных процессах в человеческой деятельности, в работе вычислительных машин и в жизни общества».

Мы будем следовать в основном последнему определению и понимать под информатикой науку об информационных процессах, методы которой находят применение в самых разных областях.

Традиционно под информатикой подразумевалась научная дисциплина, изучающая методы накопления и хранения информации (прежде всего книг и документов). К информатике также примыкают проблемы передачи информации, проектирования технических средств обучения, организации управленческой деятельности.

В средних учебных заведениях информатика и вычислительная техника изучаются в рамках одного учебного предмета, причем основное внимание уделяется вычислительной технике.

Школьный курс информатики мог бы явиться основой для унификации обозначений и терминов, методов изложения, классификации и доказательства во всех учебных курсах. Вычислительная техника может изучаться в заключительных разделах этого курса, как средство для автоматизации информационной деятельности человека. Такие понятия, как файл, каталог, иерархия, алгоритм, цикл, должны изучаться до элементов вычислительной техники. Курс логики, имеющий вековую традицию в практике отечественной школы, должен войти в курс информатики как составная часть. Важность логической подготовки учащихся осознавалась во все времена, но в современных условиях она приобретает особое значение.

Одной из задач курса информатики является развитие информационных навыков учащихся: умения составить телеграмму, разобраться в железнодорожном расписании и библиотечном каталоге, организовать свою учебу и работу, найти компромисс в споре, читать и конспектировать книгу, четко излагать свои мысли, использовать ЭВМ.

Принципиальным при построении общеобразовательного курса информатики является вопрос: учить или не учить учащихся программированию, а если учить, то на каком языке программирования?

Наиболее распространены следующие мнения по этому вопросу.

Обучать языку программирования Бейсик. Странники этой точки зрения указывают, что Бейсик имеется почти на всех ЭВМ, он получил достаточно широкое распространение во всем мире; для ведения диалога с ЭВМ пользователю достаточно знать только Бейсик, обучить ему очень легко. Противники этого языка считают, что раннее изучение Бейсика обольщивает учащихся, и их невозможно обучить в дальнейшем другим языкам программирования; в Бейсике нет естественного представления для структурных алгоритмических конструкций, программы на Бейсике трудно анализировать. Кроме того, он ориентирован исключительно на вычисления.

Обучать языку программирования Паскаль. Паскаль более структурирован, чем Бейсик. Положительный опыт массового обучения Паскалю изложен в [4].

Обучать языку программирования типа Рапира. Язык программирования Рапира (как и алгоритмический язык из пробного учебного пособия) хорошо структурирован, основан на русской лексике. Однако ориентация языка на вычисления и наличие большого числа типов данных делают проблематичным его усвоение всеми учащимися.

Обучать языку программирования Лого. Этот язык используется в ряде стран для обучения школьников младших классов. Лого легко приспосабливается к национальной лексике. Однако для старших школьников он может оказаться неинтересным, так как не имеет практических приложений.

Не обучать программированию. Программистами станет лишь небольшая часть учащихся. Для остальных знание программирования излишне, так как общеобразовательная ценность техники программирования сомнительна. Важно научить всех учащихся пользоваться готовыми программами, ведь почти каждый из них в своей будущей деятельности станет в какой-либо форме использовать ЭВМ.

Эта точка зрения имеет своих сторонников, однако такой подход не даст возможности учащимся поупражняться в конструировании алгоритмов на ЭВМ.

Возможным компромиссом могло бы стать изучение языков заданий типа используемого в операционных системах Юникс [5] и Демос [6]. Элементарными действиями этого языка являются вызовы готовых программ (с подстановкой параметров). Можно изменить ввод и вывод данных для вызываемых программ, использовать вывод одной программы в качестве ввода для другой. Кроме того, в языке заданий имеются операторы цикла, условия и выбора для построения алгоритмов из готовых программ. Эти операторы легко привести в соответствие с любой национальной лексикой. Использование языка заданий вместо языка программирования, по-видимому, облегчит конструирование алгоритмов для учащихся с наклонностями к гуманитарным предметам.

Итак, можно основные цели курса информатики свести к следующим:

- 1) ознакомление учащихся с концепциями обработки, хранения и передачи информации;
- 2) развитие мировоззрения учащихся в отношении познавательной деятельности человека, развитие логического мышления;
- 3) ознакомление учащихся с принципами управления производственными и обществен-

ными процессами, а также с информационными процессами в природе;

4) ознакомление учащихся с устройством и использованием ЭВМ.

В соответствии с этими целями предлагается включить в курс информатики следующие вопросы.

1. Формы представления информации, организации хранения, обработки и передачи информации:

формы представления информации (устные предания, наскальные надписи, рукописи, книги, журналы, микрофиши, магнитные и оптические записи), создание и использование этих форм;

организация хранения информации (архивы, библиотеки, справочные службы, делопроизводство, система научно-технической информации);

способы передачи информации (курьерская служба, почта, телеграф, телефон, радиосвязь, факсимильная связь, телевизионная связь);

кодирование информации (шифровка и расшифровка сообщений, представление чисел в различных системах счисления, помехоустойчивое кодирование);

классификация информации по ее назначению, точность и достоверность информации;

накопление информации (исследование, производство, творчество, общение);

обработка информации (копирование, запоминание, сопоставление, счет, редактирование, перевод, реферирование, принятие решений, уничтожение информации);

место ЭВМ в информационной структуре общества.

2. Использование информации для управления процессами:

сферы использования информации (управление производственными процессами, предоставление услуг, управление коллективом, обучение и воспитание, здравоохранение);

управление процессами (модели процессов, уровни детализации описания процессов, выбор действия в процессе, циклический процесс, состояние процесса); конечный автомат — простейшая модель процесса;

понятие алгоритма;

алгоритмы целенаправленной деятельности;

использование ЭВМ для управления процессами.

3. Философские и логические аспекты информатики:

постановка философских вопросов об окружающем мире;

познаваемость мира и методы познания; историческое развитие методов познания и форм мышления;

элементы логического мышления (понятие, суждение, определение, умозаключение); формализация логического мышления (операции с понятиями, основные логические законы, правила логического вывода); математические модели актов мышления и представления знаний;

логические конструкции при письменном и устном изложении;

логические и психологические особенности спора.

4. Устройство и использование ЭВМ: логические модели процессов обработки информации;

логическая структура ЭВМ; принципы физической реализации ЭВМ; устройства для хранения информации; ввод-вывод информации, линии связи; использование готовых программ для ЭВМ;

конструирование алгоритмов для ЭВМ (использование комбинаций готовых программ).

Данная заметка не ставит своей целью представить окончательные рецепты по ор-

ганизации перспективного курса информатики, а является приглашением к обсуждению.

Литература

1. *Ершов А. А. и др.* Основы информатики и вычислительной техники: Пробное учебное пособие для средних учебных заведений. Ч. I. М., 1985.

2. *Кузьмин Ю. Я.* Основы информатики и вычислительной техники. Рига, 1985.

3. *Каймин В. А. и др.* Рукопись конкурсного учебника «Основы информатики и вычислительной техники для средней общеобразовательной школы».

4. *Григас Г. К.* Начала программирования: Книга для учащихся/Пер. с лит. под ред. Ю. А. Первина. М., 1987.

5. Итоги науки и техники. Серия В. Т. 23. М.: ВИНТИ, 1985.

6. Диалоговая единая мобильная операционная система ДЕМОС: Руководство программиста. Калинин, 1985.

125

М. ПЕТРОВ

Москва

Нужен ли в спецшколе «Агат»?

«Агат» является лучшим из серийно выпускаемых ныне в СССР школьных компьютеров и для преподавания ОИВТ в средней школе вполне приемлем.

А теперь представим себе следующую ситуацию: физико-математическая школа, в которой ведется профессиональное обучение по специальности «Оператор электронно-вычислительных машин», получает в свое распоряжение ПЭВМ «Агат». Реализовать курс, рассчитанный на СМ или ЕС ЭВМ на персональном компьютере практически невозможно, и школьников, в обход учебного плана, начинают учить программированию. Учить серьезно. «Агат» для этой цели не подходит, и вот из каких соображений:

1. Учащиеся математических классов, как правило, не удовлетворяются Бейсиком, а другого языка высокого уровня нет (если не считать доморощенную Рапиру или экзотический Форт). Следовательно, остается одно — язык ассемблера. Но, пожалуйста, любое обучение должно иметь практическую направленность, а система команд процессора 6502 несовместима с теми ассемблерами, что ждут ученика в дальнейшем (все-

таки значительная часть школьников будет связана с ВТ).

2. Прежде чем переходить к программированию в машинных кодах, необходимо научиться писать хорошие программы на нескольких языках высокого уровня, таких, как Паскаль или Си, а их то и нет.

3. Помимо программирования ученик должен приобрести навыки работы с разнообразным программным обеспечением, которого опять же практически нет. Этот пункт не менее важен, чем все предыдущие, так как без развитого обеспечения приходится замыкаться на программировании, что губительно для информатики.

У читателя может возникнуть впечатление, что все сказанное относится и к обычной, средней школе. Однако если там урок информатики — 1—2 ч в неделю, то, повторюсь, «Агат» вполне приемлем. Но когда речь идет о 5 ч в неделю...

Думается, что и у пользователей КУВТ-86 найдутся возражения против присутствия одного в математических школах. В этих школах нужна иная техника, более надежная, с более развитым обеспечением.

Новый центр информатики

короткая экскурсия в 2 частях, в первой из которых автор не пожалеет розовой

краски, а во второй приоткроет окно, чтобы подуло холодным ветерком.

Итак, мы в новом центре информатики. Он в Москве, на Смоленском бульваре, в доме № 4, открылся 19 сентября, когда столица праздновала День города. Новорожденному исполнилось несколько месяцев, и поэтому в поздравительно-информационной части позволю себе процитировать директора центра В. Бусленко (из неопубликованного!):

*Приводи, бабуля, внука
И сестренка — братика.
Отдых, польза и наука —
Наша информатика.*

В компьютерном центре — зал для профессионального программирования, видеотека (знай наших!) с буфетом или буфет с видеотекой и, к неподдельному восторгу малолетних, игровой зал. Все услуги платные, но цены вполне по карману людям с трудовыми доходами. Вскоре такие центры нач-

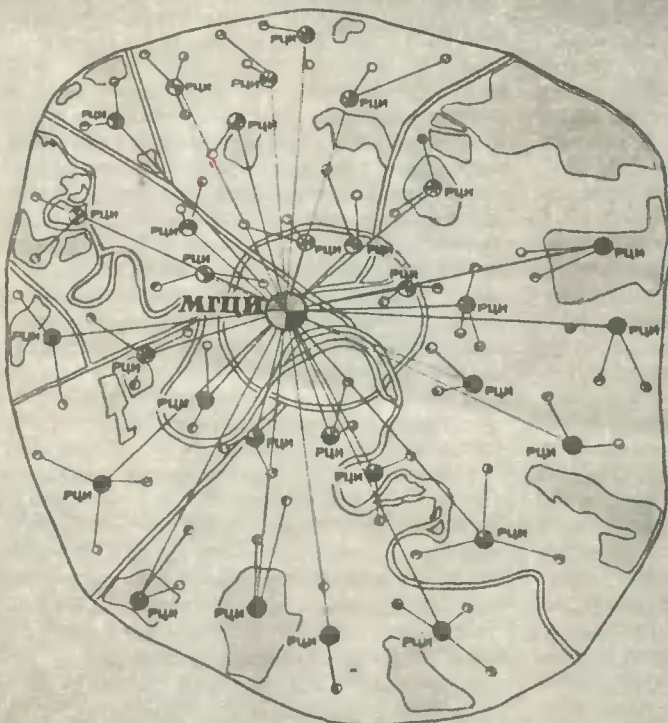
нут появляться, как грибы после дождя (рис. 1). Это отраднo. А поскольку мы путешествуем по одному из первых, оптимизм наполняет душу и воображение рисует, что счастливые дети поют:

*Что мне шашки, домино,
Куклы и солдатики —
Интересней все равно
В центре информатики!*
(В. Бусленко)

Все это так: есть и телефон, и директор, есть справочно-информационный отдел, временный прейскурант и много-много очень дорогих вычислительных машин. Нет пока дружественного программного обеспечения (что мы подразумеваем под этим понятием, вы узнаете из наших последующих публикаций); нет прибыли и нет правового статуса, который бы определил, что есть центр

126

СЕТЬ ЦЕНТРОВ ИНФОРМАТИКИ г.МОСКВЫ проект



*Адрес центра
111034 Москва
Смоленский бульвар 4.*

*Телефон центра
246-96-59*



РЕПЕРТАЖ
НОМЕРА



и чем он отличается от магазина, библиотеки или бани, то бишь от торгового, культурно-просветительного учреждения или предприятия сферы обслуживания (а центр отличается от того, другого и третьего).

Если начать втискивать новое дело в привычные устоявшиеся рамки, может и в новом деле приключиться застой. Да и не только в этом суть. Когда нет прецедентов, надо создавать новые формы, а не подгонять новое под старое.

Общество компьютеризируется. Люди хотят получить «вторую» грамотность. Но надо, простите за каламбур, и грамотно дать им ее.

Поэтому, чтобы столь блестящее по замыслу дело не превратилось в средство для успокоения комиссий («и нас не миновал прогресс!»), необходимо помочь центру встать на ноги.

Игровой зал всегда заполнен желающими развлечься. Это, конечно, ничего общего не имеет с изучением информатики (симптомы «детской болезни» описаны психологом А. Шмелевым*). Симпатичные ребята (рис. 2), обслуживающие этот зал, изучали информатику в школе, но вряд ли им хватит знаний и квалификации (за плечами только десять классов) объяснить «изюминку»

каждой программы. Так что не будем обманывать себя — у центра много проблем. Пожелаем его руководству терпеливо и мудро растить кадры, помогать молодым обрести себя.

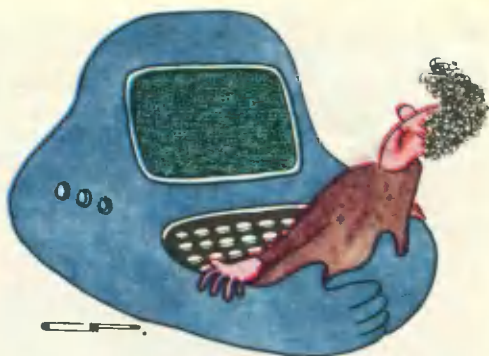
Газетные публикации о центре поверхностны и несколько сенсационны. Чего стоят одни заголовки, например: «Компьютер на хозрасчете». Ни о каком хозрасчете, в его подлинном смысле, пока нет и речи. Те, кто мог бы зарабатывать здесь по труду (в центре, как и везде, есть специалисты и энтузиасты), получают зарплату, несоизмеримую с отдачей. Не работает принцип материальной заинтересованности. На зарплате работников не отражается — 70 или 1070 рублей прибыли в день принесет центр. Обществу центр пока дает копейки (и по этой причине тоже). Материальная сторона не главное в битве за умы, скажете вы. Да, это так. Однако по имеющемуся программному обеспечению видно, кто борется за наши умы (профессионалы поймут, что имеется в виду).

Реальную помощь можно было бы оказать, предоставив центру прежде всего свободу действий, организовав умную и достоверную рекламу. (А то из публикаций следует, что компьютер чуть ли не домой можно взять: пока этого нет.) И конечно, рано успокаиваться и рапортовать.

Поэтому мы еще вернемся сюда.

Экскурсию провели Т. ДРАГНЬШ

* Информатика и образование, 1987, № 1. Автор ставит вопрос об ответственности организаторов досуговых центров.



ЗНАЕТЕ ЛИ ВЫ, ЧТО...

... установка персонального компьютера еще не подразумевает выделение персонального программиста.

...использование дисковода в качестве обеденного столика нерационально из-за раздражающей вибрации устройства.

...одновременное нажатие пяти-шести клавиш на клавиатуре работающего компьютера дает иногда более неожиданный результат, чем нажатие одной-двух.

...потребление электроэнергии микрокомпьютером и цветным телевизором различается незначительно, но польза от последнего во время трансляции спортивных программ неизмеримо выше.

...еще никто не пробовал включать персональную ЭВМ в трансляционную сеть.

...компьютер, поставленный в угол, далеко не всегда чувствует себя наказанным.

...запрещается пользоваться аспирином, малиновым вареньем и горчичками для снижения температуры перегревшегося компьютера.

КОМПЬЮТЕРЫ И ПЕРСОНАЛИИ

«...Обеспечим каждого представителя технического персонала персональной техникой, а каждую единицу персональной техники необходимым для ее ремонта техническим персоналом, который, в свою очередь, обеспечим...»

НЕМНОГО О МПК

«...На ваш запрос об ускорении поставки микропроцессорного комплекса (МПК) сообщаем, что микропроцессор уже отгружен в адрес заказчика товарной платформой № 150278, что же касается комплекса, то его отправка задерживается по техническим причинам до конца следующего года».

ИЗ ПРИКАЗА...

«...В связи с окончанием гр. Сидоровым И. И. курсов по ликвидации компьютерной безграмотности переименовать занимаемую им должность переводчика в должность транслятора. Основание: заявление гр. Сидорова».

ОБЪЯВЛЕНИЯ

— Меняю одну двухпроцессорную ЭВМ на две однопроцессорные в разных аудиториях.

— Интерпретирую программы и двигаю курсор...

— Приглашаю няню-программиста к компьютеру двух лет.

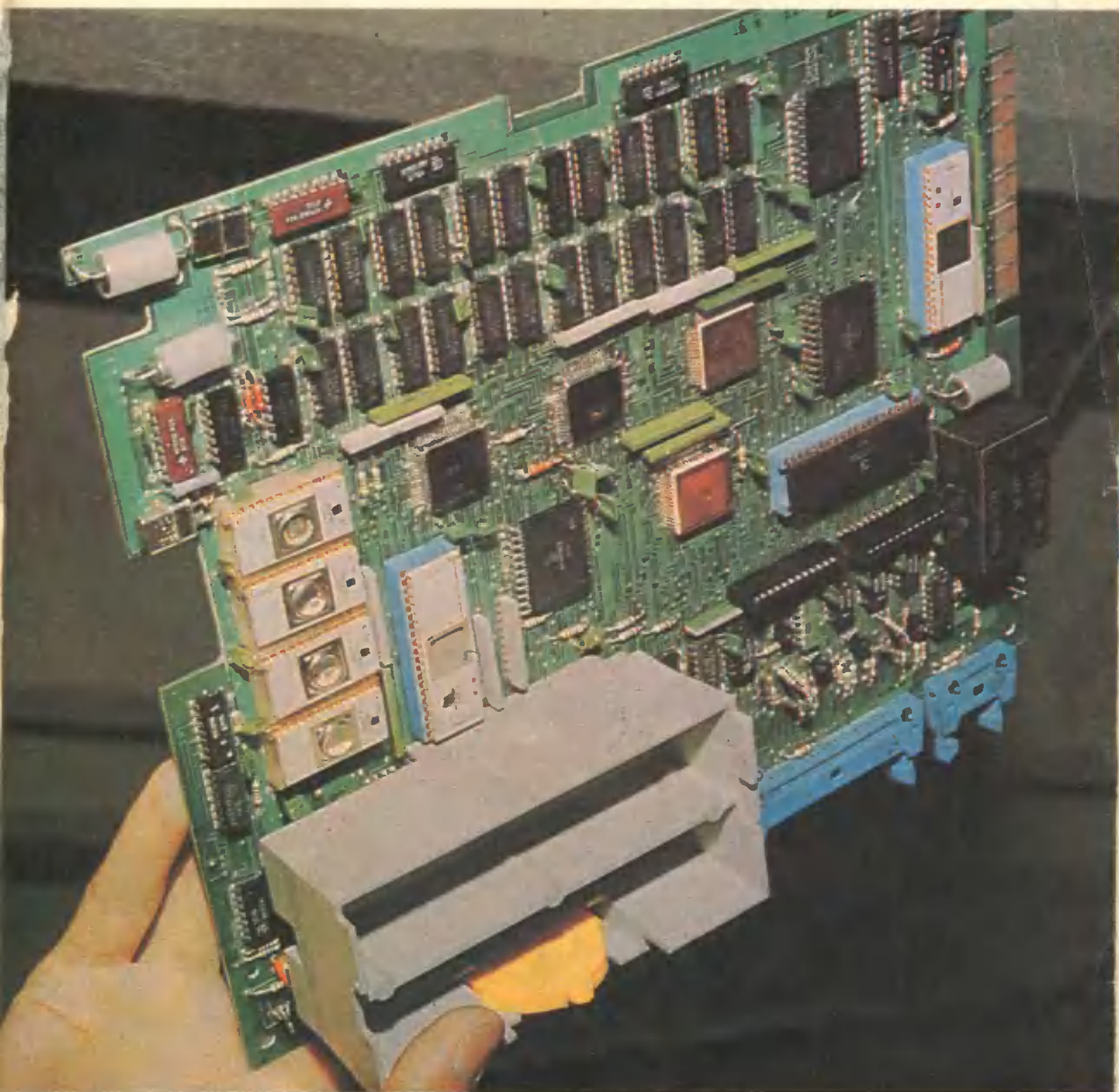
МЫСЛИ И ФРАЗЫ...

— А все-таки: в какой системе счисления лучше получать зарплату?

— Из сборника сказок для программистов младшего возраста: «Давным-давно, когда компьютеры были большими...»

— ЭВМ и мода: макси, миди, мини... !

Ю. ШЕХОНИН



**Графические возможности новой ЭВМ сегодня и в перспективе
(1-я и 4-я страницы обложки): теперь дело за программистами**

Цена 60 коп.

083

70423

ИНФО
1'88



**ИНФОРМАТИКА
И ОБРАЗОВАНИЕ**

