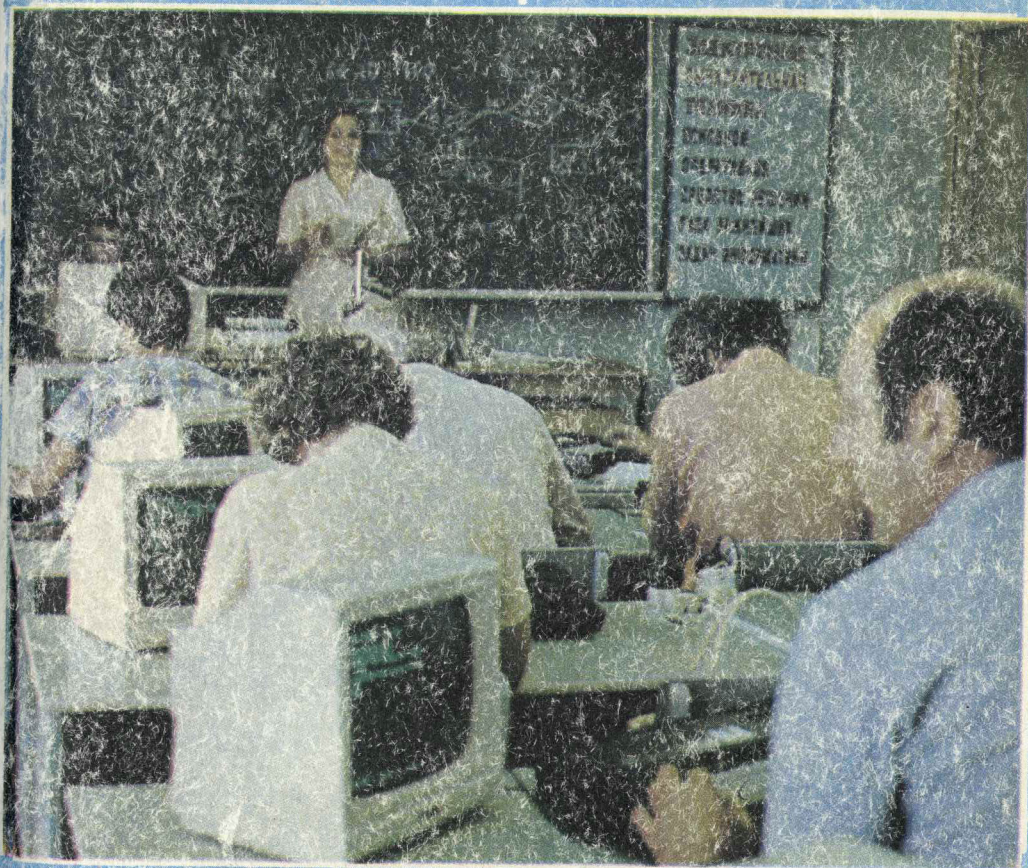


# ИНФОРМАТИКА И ОБРАЗОВАНИЕ

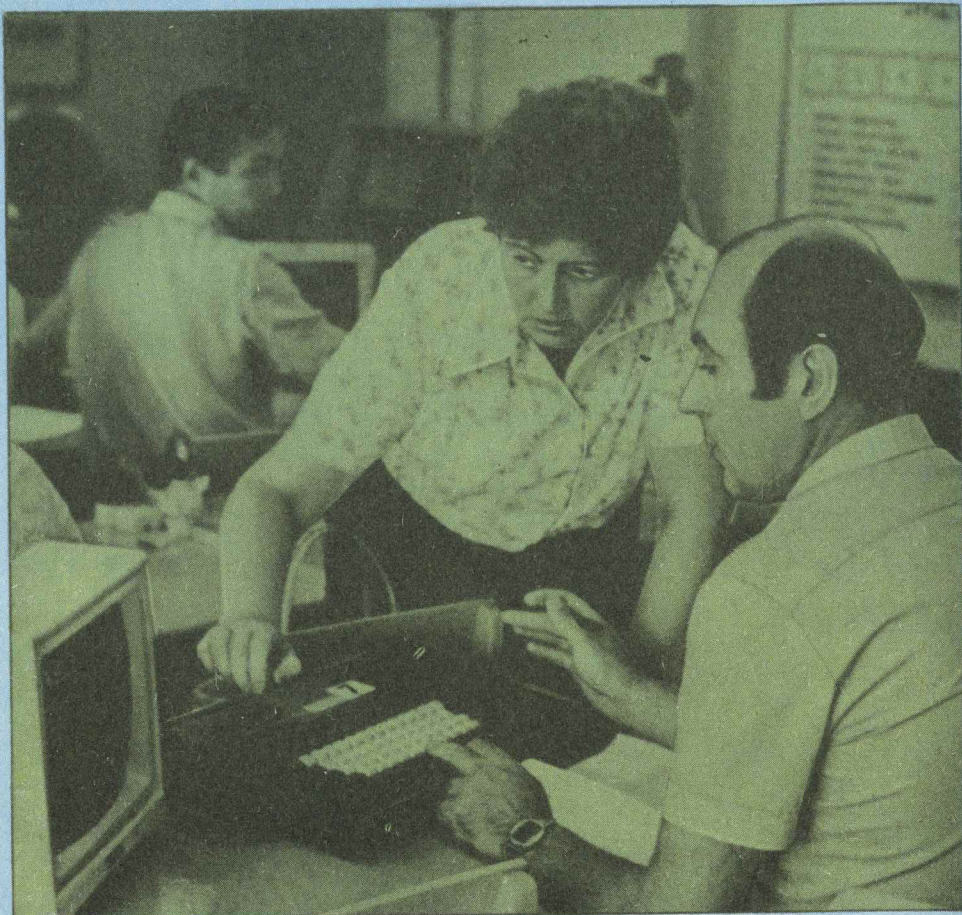
3 1986

## В НОМЕРЕ:

- Педагоги за дисплеем
- Калькулятор БЗ-34
- Бейсик для всех
- «Черепаша» учит программировать







## Педагоги за дисплеем

Летом многие педагоги познакомились с вычислительной техникой на курсах переподготовки. Они были организованы и в Московском областном институте усовершенствования учителей (МОИУУ) (фото на 1-й странице обложки). Занятия проводились по 36-часовой программе для учителей математики, физики, химии и по 12-часовой программе — для преподавателей других предметов. Половина учебного времени отводилась работе на ЭВМ. Учителя преодолели своеобразный психологический барьер, убедились, что вести диалог с компьютером может каждый. Интерес к занятиям был чрезвычайно велик. В дисплейном классе педагоги не только познакомились с современной вычислительной техникой, но и сами составляли несложные программы по своему предмету. Обучающие программы, по которым готовились педагоги, были составлены не только по информатике, физике, математике, но и по биологии.

Чтобы повысить эффективность обучения педагогов в МОИУУ, оборудуются новые дисплейные классы на базе отечественной техники (КУВТ-86).

Подробно о летних курсах учителей, организованных в МОИУУ, рассказывается в статье М. Ратинского, опубликованной в этом номере журнала.

Научно-методический журнал  
Министерства просвещения СССР,  
Государственного комитета СССР  
по профессионально-техническому образованию  
Министерства высшего и среднего  
специального образования СССР

Выходит шесть раз в год  
Издается с 1986 г.

Издательство «Педагогика»  
Москва

**ИНФ**  
**3 '86**  
ноябрь —  
декабрь

# ИНФОРМАТИКА И ОБРАЗОВАНИЕ

1

## Содержание

Встреча с читателями 3

## В Министерстве просвещения СССР

Кабинет вычислительной техники всех типов средних учебных заведений  
(на базе персональных микро-ЭВМ). Методические рекомендации 5

## Общие вопросы

Кобринский Я., Кузнецов А. Особенности пакетов прикладных программ 18  
Баринев А. Компьютер в ПТУ: сегодня и завтра 25  
Волков Д., Ширков П., Деянов Р. Опыт ускоренного обучения 28

## Методика обучения

Методические указания по преподаванию курса «Основы информатики  
и вычислительной техники» в X классе 31

## Кабинет ВТ

Ершов А. Решение задач с применением программируемого микрокаль-  
кулятора «Электроника БЗ-34» 61  
Икаунекс Е. Основы языка Бейсик 73  
Леонова Л., Савватеева С. Занятия в КВТ: влияние на организм 82  
Викентьев Л., Козлов О., Костров А. ЕС ЭВМ в курсе информатики 82

## Внеклассная работа

Туманова О., Лихачева Л. «Иначе бы мы к вам не пришли...» 85  
Антипов И. Управляемый карандаш 87  
Флеров М. «Черепашка» учит программировать 94  
Пачиков С. Дома за дисплеем 96

## ЭВМ в народном хозяйстве

Поляков В. На суше и на море 101



## Репортаж номера

Латвийский опыт: вуз помогает школе

## Информация

## Педагогические кадры

Ратинский М. Курсы для учителей

## Нам пишут

Тутаринов Ю., Чиняев А. «Пропагандист»  
Славин Д. Каждый урок — эксперимент  
Юшков С., Андрюхагин А. Проблемы нового курса

## Веселый урок

108

84

110

114

115

120

123

## Встреча с читателями

Наш первый номер вышел в свет к началу нового учебного года. Редколлегия и редакция журнала, его издатели и авторы с нетерпением ожидали отзывов читателей о журнале. Встреча с ними состоялась на выездном заседании редколлегии в субботу 20 сентября в Московском областном институте усовершенствования учителей, администрация которого взяла на себя все организационные хлопоты, включая приглашение педагогов и методистов области. Заседание открыл директор МОИУУ Э. М. Никитин. Он представил гостям редакционную коллегию и подчеркнул, что работники народного образования очень ждали выхода в свет такого журнала, который станет научно-методической основой становления нового предмета во всех звеньях народного образования в соответствии с решениями XXVII съезда КПСС.

Тепло был встречен присутствующими главный редактор журнала — директор Института проблем кибернетики АН СССР академик В. А. Мельников. Он рассказал об истории становления отечественной школы вычислительной техники, спорах ученых вокруг предмета и методов современной информатики, центральной частью которой является компьютер и вычислительные науки. И хотя пока не совсем ясно, как надо преподавать информатику, очевидно одно — необходимо развивать логическое мышление, владение новым инстру-

ментом, рассматривая предмет как общеобразовательный, не увлекаясь программированием.

От имени учителей области выступила О. К. Павлова — учитель школы из г. Орехово-Зуево и Н. Г. Мельдианов — учитель из поселка «Заветы Ильича». По их мнению, журнал не должен сворачивать с избранного пути, оставаясь именно научно-методическим, сохраняя ориентацию прежде всего на школьного учителя, отстаивая его интересы. Материалы первого номера полезны и необходимы учителю для постоянного пользования. Оперативно отреагировали Минпрос РСФСР, Минвуз и Госпрофобр, централизованно закупив значительную часть тиража первого номера для его распространения в школах, техникумах, профтехучилищах. Во многих районах Московской области на августовских чтениях учителя получили в подарок первый номер журнала.

По мнению А. С. Земблинова — преподавателя СПТУ № 34 Москвы, научно-методическая основа журнала не должна препятствовать появлению отдельных популярных материалов.

В ряде выступлений (К. Н. Курбатова — зав. кафедрой информатики Института народного хозяйства им. Плеханова, Р. Н. Иванова — преподавателя института повышения квалификации работников печати, проф. К. В. Тараканова — зав. кафедрой информатики Московского института культуры,

Главный редактор  
академик  
В. А. МЕЛЬНИКОВ

Редакционная  
коллегия

И. М. БОБКО  
Б. М. ГЕРАСИМОВ  
Ф. В. ДАНИЛОВСКИЙ  
(и. о. зам. главного редактора)  
А. В. ДЕНИСЕНКО  
А. П. ЕРШОВ  
С. А. ЖДАНОВ  
Б. Ф. ЛОМОВ  
Н. Г. МЕЛЬДИАНОВ  
О. К. ПАВЛОВА  
А. Ю. УВАРОВ  
А. И. ФУРСЕНКО  
В. О. ХОРОШИЛОВ

Редактор отдела  
К. ШЕХОВЦЕВ  
Научный редактор  
Т. ДРАГНЫШ

Заведующая редакцией Н. Игнатова  
Технический редактор Л. Розанова  
Корректор Н. В. Минервина

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

Почтовый адрес: 107847 ГСП, Москва, Б-66, Лефортовский пер., д. 8

Сдано в набор 01.10.86. Подписано в печать 03.11.86. А 08738. Формат 70×100/16. Печать офсетная. Усл. печ. л. 10,4. Уч.-изд. л. 13,64. Усл. кр.-отт. 42,88. Тираж 25460 экз. Заказ 2730. 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 обл.

© Издательство «Педагогика» «Информатика и образование», 1986



Б. М. Герасимова — члена редколлегии журнала) звучало настойчивое мнение о необходимости более расширенного толкования предмета информатики в журнале с включением в него так называемой традиционной информатики, не привязанной к компьютеру как таковому. Подрастающее поколение должно уметь обращаться с микрофишами и архивами, каналами связи и банками данных, владеть современной культурой и дисциплиной общественных коммуникаций. Нам не надо добиваться массового производства дилетантов от программирования.

Научный сотрудник Института психологии АН СССР А. В. Дрынков рассказал о планах журнала по освещению психолого-педагогических проблем компьютеризации образования.

О роли молодежи в решении проблем компьютеризации на примере внеклассной и внешкольной работы рассказал инструктор ЦК ВЛКСМ А. И. Парамонов.

Считаясь с реальным положением дел на местах, журнал должен уделять внимание и работе с калькуляторами — это отметил Б. А. Тарасенко.

От научной общественности выступили Ю. А. Первин (г. Переславль-Залесский) и К. М. Шоломий, которые высказали конкретные пожелания журналу в части описания обучающих программ, работы со школьниками.

На встрече была высказана озабоченность педагогов произошедшим в последний год креном в сторону импортной техники, а серийные модели отечественных компьютеров оказались вне поля зрения организаций, ответственных за разработку программного и методического материала. Журнал должен стараться компенсировать подобные явления.

Подчеркивалась большая воспитательная роль статьи В. А. Мельникова об основоположнике отечественной вычислительной техники С. А. Лебедеве. Такие статьи способствуют патриотическому воспитанию школьников.

Учителям были представлены сотрудники редакции Н. В. Игнатова, Т. Н. Драгныш, К. В. Шеховцев, обеспечившие своевременный выпуск первых номеров журнала в сложных условиях.

В заключение выступили представители издателей — зам. начальника Управления информатики Минпроса СССР А. Н. Ковалев, начальник отдела информатики Госпрофобра СССР В. О. Хорошилов, представитель Минвуза СССР А. А. Коптев, которые рассказали о планах ведомств системы народного образования в отношении нового журнала.

Редколлегия журнала и учителя области признательны администрации МОИУУ за проведенную встречу.

*Дорогие читатели!*

*Редколлегия и редакция поздравляют Вас  
с наступающим Новым годом,  
желают здоровья, счастья, больших творческих успехов!  
Уверены, что в 1987 году  
журнал «Информатика и образование»  
будет Вашим хорошим помощником в педагогической работе,  
знакомстве с современными ЭВМ.  
Ждем от Вас интересных писем и предложений.*

## В Министерстве просвещения СССР

### Кабинет вычислительной техники всех типов средних учебных заведений (на базе персональных микро-ЭВМ). Методические рекомендации\*

#### Введение

Методические рекомендации по оборудованию кабинета вычислительной техники разработаны согласно постановлению апрельского (1984 г.) Пленума ЦК КПСС «Об основных направлениях реформы общеобразовательной и профессиональной школы» и постановлениями ЦК КПСС и Совета Министров СССР.

Источниками для разработки рекомендаций послужили: «Заключение межведомственной экспертной группы по отбору ЭВМ и оборудования для кабинетов информатики и вычислительной техники для общеобразовательной и профессиональной школы» от 1.02.85 г., одобренное министром просвещения СССР; учебная программа и пробное учебное пособие по курсу «Основы информатики и вычислительной техники»; опыт оборудования кабинетов по общеобразовательным и другим предметам, накопленный школами и средними профессиональными учебными заведениями за последние годы; опыт создания кабинетов вычислительной техники в средних учебных заведениях Москвы, Новосибирска и других городов; опыт применения ЭВМ в школах стран социалистического содружества. Использован также (критически) опыт применения компьютеров в школах капиталистических стран.

#### 1. Общие положения

##### 1.1. Кабинет вычислительной техники

\* Разработаны в НИИ ШОТСО АПН СССР под руководством С. Г. Шаповаленко и Е. П. Смирнова. В составлении рекомендаций принимали участие Я. В. Владимиров, Н. Л. Максимова, И. В. Роберт, Е. П. Смирнов, В. П. Царев, И. М. Яглом (НИИ ШОТСО), Е. А. Гельтищева, В. М. Кирихин (МНИИ гигиены им. Ф. Ф. Эрисмана).

(КВТ) — это учебно-воспитательное подразделение средней общеобразовательной и профессиональной школы, межшкольного учебно-производственного комбината, оснащенное комплексом учебной вычислительной техники (КУВТ), учебно-наглядными пособиями, учебным оборудованием, мебелью, оргтехникой и приспособлениями для проведения теоретических и практических занятий по курсу «Основы информатики и вычислительной техники» (ОИВТ). КВТ используется в преподавании различных учебных предметов, трудового обучения, в организации общественно полезного и производственного труда учащихся, для эффективного управления учебно-воспитательным процессом.

##### 1.2. Занятия в КВТ должны служить:

— формированию у учащихся компьютерной грамотности — знаний об устройстве и функционировании современной вычислительной техники, умений и навыков решения задач с помощью ЭВМ;

— ознакомлению учащихся с применением вычислительной техники на производстве, в проектно-конструкторских организациях, научных учреждениях, учебном процессе и управлении;

— развитию у учащихся теоретического мышления в процессе изучения основ информатики;

— совершенствованию методов обучения и организации учебно-воспитательного процесса в школе;

— формированию личности советской молодежи этапа планомерного и всестороннего совершенствования социализма.

##### 1.3. В КВТ проводятся:

— занятия по ОИВТ и отдельным общеобразовательным учебным предметам с использованием электронно-вычислительной



техники, кинофильмов, диапозитивов, таблиц и других учебно-наглядных пособий;

— составление учащимися прикладных программ по заданиям учителей и руководства школы для удовлетворения потребностей школы и базовых предприятий (работы учащихся по составлению программ проводятся с использованием магнитных носителей);

— внеклассные и факультативные занятия по ОИВТ;

— экспериментальные уроки и практические занятия.

1.4. КВТ может быть школьным (обслуживать одну школу) или межшкольным (обслуживать учащихся нескольких школ, училищ). Число рабочих мест для учащихся может быть 9, 12, в зависимости от наполняемости классов. На рабочем месте предусматривается работа одного-двух учащихся<sup>1</sup>.

Для проведения практических занятий с компьютерами классы делятся на две подгруппы (в городских школах — классы с количеством учащихся 25 и более человек, а в сельских школах — 20 и более человек).

1.5. КВТ должен быть выполнен как психологически, гигиенически и эргономически комфортная среда, организованная так, чтобы в максимальной степени содействовать успешному преподаванию, умственному развитию и идейно-политическому воспитанию учащихся, приобретению ими прочных знаний, умений и навыков по ОИВТ и основам наук, при полном обеспечении требований к охране здоровья и безопасности труда учителя и учащихся.

1.6. В КВТ должно быть обеспечено информационное взаимодействие между учащимися и техническими средствами хранения и обработки информации, между учащимися и учителем, необходимое для осуществления учебно-воспитательного процесса.

## 2. Оснащение КВТ учебным оборудованием, учебно-наглядными пособиями, оргтехникой и мебелью

2.1. Для реализации задач и содержания работ, предусматриваемых пп. 1.2 и 1.3, КВТ оснащается материальными средствами, согласно «Перечням технических средств, учебно-наглядных пособий и мебели для кабинетов вычислительной техники всех типов учебных заведений». Приложение 1.

2.2. Кроме того, КВТ оснащается:

— набором учебных программ по ОИВТ и отдельным разделам учебных предметов (математика, физика, химия, география, иностранный язык и др.);

— заданиями (по возможности записанными на магнитных носителях) для осуществления индивидуального подхода при обучении, организации самостоятельных работ и упражнений учащихся на компьютерах;

— комплектом научно-популярной, справочной и методической литературы;

— набором лучших программ для ЭВМ, созданных учащимися;

— журналом вводного и периодического инструктажей учащихся по технике безопасности (Приложение 5, п. 2);

— журналами использования КУВТ на каждом рабочем месте;

— журналом отказов машин и их ремонта;

— держателями для демонстрации таблиц и стендами для экспонирования работ учащихся;

— инвентарной книгой для учета имеющегося в кабинете учебного оборудования, годовыми и пятилетними планами дооборудования КВТ, утвержденными директором школы;

— аптечкой первой помощи (Приложение 6, табл. 3);

— огнетушителем.

2.3. Учебное оборудование и учебно-наглядные пособия для КВТ приобретаются школами через магазины Главснабпросов в установленном порядке в соответствии с «Перечнями технических средств, учебно-наглядных пособий и мебели для кабинетов вычислительной техники всех типов учебных заведений» (Приложение 1)<sup>2</sup>.

## 3. Оборудование рабочих мест учащихся и учителя

3.1. Рабочие места учащихся, оснащенные персональными ЭВМ (ПЭВМ), должны состоять из двухместных столов со стульями, разработанными с учетом ГОСТ 11015-77 «Столы ученические. Типы и функциональные размеры» и ГОСТ 11016-77 «Стулья ученические. Типы и функциональные размеры». Размер крышки стола должен быть 1200×700 мм, чтобы разместить ПЭВМ, учебник и иметь площадь для записи работы с книгой, высота — 660 и 720 мм (соответственно группам В и Г) (Приложение 2). Столы и стулья для учащихся подбираются

по группам В и Г в отношении 1:1.

3.2. На столах устанавливаются ПЭВМ со всеми необходимыми периферийными устройствами. К столам подводятся электропитание и кабель локальной сети. Столы оборудуются в соответствии с требованиями безопасности и крепятся к полу.

3.3. Рабочее место учителя оборудуется столом, оснащенным аппаратурой в соответствии с «Перечнем» (Приложение 1) и двумя тумбами для принтера и графопроектора. Размеры стола учителя: длина крышки не менее 1200 мм, ширина не менее 700 мм, высота 720 мм. В тумбах предусматривается 1—2 ящика размерами не менее 350×500×100 мм для принадлежностей, магнитных носителей и транспарантов из расчета на текущий день занятий. Стол используется для установки ПЭВМ и ведения записей. Он может иметь нишу для тетрадей, классного журнала и т. п. (Приложение 3).

3.4. При оснащении кабинета микро-ЭВМ, в которых клавиатура совмещена с системным блоком, рекомендуется подставка под монитор на ученический стол (Приложение 2а).

## 4. Помещение для КВТ, размещение рабочих мест, хранение учебного оборудования

4.1. Примерное размещение рабочих мест учащихся и учителя в КВТ приведено в приложениях 4а, 4б, 4в, 4г. Это размещение исключает, согласно правилам техники безопасности (Приложение 6), соприкосновение учащихся с задними стенками мониторов (телевизоров).

Для обеспечения этого используются также такие меры, как размещение мониторов задними панелями к стене, устройство задних перегородок и т. д.

Вариант 4а расположения рабочих мест учащихся и учителя в КВТ (Приложение 4) рассчитан на 18 рабочих мест учащихся и 9 ЭВМ. Вариант 4б рассчитан на 24 рабочих места учащихся и 12 ЭВМ. Он выполнен в традиционном стиле расположения рабочих мест учащихся и учителя.

Вариант 4в, рассчитанный на 30 рабочих мест учащихся и 12 ЭВМ, предполагает следующую форму работы в КВТ: те учащиеся, которые заняты подготовительной или контрольной работой (не заняты непосредственно работой на ЭВМ), располагаются за столами (1.1), расположенными в центре кабинета. Это, во-первых, не отвлекает ученика, занятого письменной работой, к ЭВМ и, во-вторых, позволяет его напарнику спокойно работать у ЭВМ. В случае

коллективной работы все учащиеся рассаживаются парами за каждой ЭВМ. Вариант 4г, рассчитанный на 32 рабочих места учащихся и 9 ЭВМ, повторяет преимущество варианта 4в. Кроме того, этот случай учитывает разделение учащихся на две подгруппы, максимум по 16 человек в каждой (при 9 ЭВМ).

Общая электрическая схема питания для КВТ включается в сопроводительную документацию, поставляемую с комплектом электрооборудования кабинета вычислительной техники (КЭВТ) (Приложение 1).

4.2. Передняя стена КВТ оборудуется классной доской, экраном, шкафом для хранения учебно-наглядных пособий и носителей информации и демонстрационным монитором (телевизором)<sup>3</sup>.

Под доской устанавливают два ящика для таблиц. На верхней кромке доски укрепляются держатели (или планка с держателями) для подвешивания таблиц.

4.3. Учебные пособия и оборудование размещаются и хранятся по разделам программы в кабинете. Демонстрационные пособия и оборудование для самостоятельных работ хранятся раздельно.

4.4. Для хранения учебно-наглядных пособий и оборудования КВТ оснащается шкафом, устанавливаемым справа от классной доски.

4.5. Демонстрационные пособия хранятся в КВТ следующим образом:

— таблицы и репродукции — в ящиках под доской или в специальных отделениях шкафов по разделам программ и классам с учетом габаритов;

— аудиовизуальные пособия — записи на магнитных лентах — хранятся на полках шкафов, диафильмы и диапозитивы — в укладках с выемками для коробок;

— диски (кассеты) с программным материалом — в специальном небольшом ящике, размером не менее 150×150×300 мм, защищенном от пыли и света, по классам и разделам программы. Ящик помещается в шкаф, а места хранения в нем дисков отмечаются надписями;

— справочная, методическая и научно-популярная литература — на полках шкафа (поставляется школьной библиотекой).

4.6. В КВТ создается картотека имеющегося учебного оборудования с указанием мест хранения (карточки хранятся в алфавитном порядке) и методическая картотека, облегчающая учителю и лаборанту подготовку оборудования к занятиям.

4.7. На стене, противоположной окнам,

<sup>3</sup> Демонстрационный телевизор устанавливается на высоте 2 м от пола на кронштейне.

<sup>1</sup> Положение о межшкольном КВТ и рекомендации по организации работы в нем будут публиковаться в виде отдельного документа.

<sup>2</sup> Профтехучилища получают дополнительные средства обучения через Всесоюзное объединение производственных предприятий Госпрофобра СССР.



размещаются щиты с постоянно находящимися в кабинете справочными таблицами, знаками учащихся с правилами по технике безопасности, основными узлами ЭВМ и их функциями, видами алгоритмов и т. д.

4.8. Пособия, необходимые для изучения отдельных тем и разделов программ, экспонируются в КВТ на задней стене.

Для экспозиции пособий, книг и материалов КВТ оснащается съемными стендами 90×90 см. Экспозиция устанавливается по наиболее важным или трудным темам курса, а также по темам, по которым учащиеся провели большую самостоятельную работу. При переходе к изучению другой темы материалы экспозиции предыдущей темы заменяются новыми.

Следует предупреждать перегрузку кабинета стендами с указанными материалами. Часть материалов может быть вынесена в коридор на стенды перед входом в КВТ.

4.9. В КВТ используется комплект технических средств обучения (ТСО), имеющийся в школе (Приложение 1).

## 5. Гигиенические рекомендации к организации КВТ

5.1. Во время работы учащихся на ПЭВМ в кабинете повышается температура и снижается относительная влажность воздуха, ухудшается ионный и качественный состав воздуха: увеличивается содержание в воздухе органических веществ и двуокиси углерода. Содержание в воздухе указанных веществ может в несколько раз превышать предельно допустимые величины.

5.2. В школах, выстроенных по проектам, в которых не предусматривались КВТ, для поддержания надлежащего микроклимата (оптимальная температура воздуха в КВТ для школ средней полосы должна находиться в пределах 19—21 °С, а для школ, расположенных на севере и юге, в пределах 20—22 °С; относительная влажность — не менее 30—45 %) и улучшения качественного состава воздуха, в том числе и ионного, необходимо перед учебными занятиями, а если позволяют погодные условия, то во время учебных занятий и перемен хорошо проветривать помещение кабинета через открытые окна, фрамуги. Последние должны находиться в верхней части окна и открываться под углом 30—40° в сторону учебного помещения. При этом механическая регулировка фрамуг без дополнительных устройств (палок с крючками и т. п.) должна осуществляться на уровне 1,2—1,4 м от пола.

Проветривание КВТ через открытые окна и фрамуги из-за поступления пыли вместе

с воздухом может снижать надежность работы ЭВМ.

На воздухообмен в КВТ оказывает влияние площадь и кубатура КВТ. Минимальная площадь, приходящаяся на одну машину, должна быть не менее 5 м<sup>2</sup>. Зона перед классной доской, где располагается рабочий стол учителя и демонстрационный телевизор, по длине кабинета должна быть не менее 2,5 м. Высота КВТ — 4 м.

При КВТ необходимо иметь подсобное помещение (20 м<sup>2</sup>).

Для исключения влияния солнечной радиации на микроклимат и создания равномерного естественного освещения КВТ следует ориентировать на северную сторону горизонта.

При невозможности организации КВТ с оконными проемами на северную сторону горизонта необходимо предусмотреть для окон солнцезащитные регулируемые устройства типа жалюзи, расположенные снаружи или в межстекольном пространстве.

Для улучшения воздухообмена и снижения температуры воздуха можно рекомендовать установку бытовых кондиционеров типа БВ-1500, БК-2500.

Микроклимат зависит также и от типа ЭВМ, количества людей и других факторов, поэтому в каждом конкретном случае при организации КВТ необходим расчет воздухообмена, который должен проводиться с учетом разбавления теплоизбытков от мониторов и людей, а в случае ориентации окон на другие стороны горизонта необходимо учитывать тепло, создаваемое солнечной радиацией.

5.3. Для снижения шума при работе кондиционеров можно рекомендовать «шумо-заглушающее устройство для автономного кондиционера, направляющее воздушный поток», разработанное МНИИ гигиены имени Ф. Ф. Эрисмана и ЦНИИ экспериментального проектирования учебных зданий, а также отделку потолка или стен выше панелей (1,5—1,7 м) звукопоглощающими материалами типа «акмигран», «акминит», плитами марки ПА/О, ПА/С.

Окрашивать звукопоглощающие плиты не разрешается!

5.4. Пол должен иметь «Поливинилхлоридное двуслойное антистатическое покрытие для съемных полов» (ТУ 21-29-108-84).

Двери и стенные шкафы могут быть облицованы «Поливинилхлоридным декоративным антистатическим материалом» (ТУ 400-20-38-3-82).

В КВТ в целях снижения загрязнения воздуха вредными химическими веществами запрещается для отделки интерьера и оборудования применять строительные мате-

риалы, содержащие органическое сырье: древесностружечные плиты (ДСП), декоративный бумажнослоистый пластик (ГОСТ 9590-76), поливинилхлоридные пленки и др.

5.5. Освещенность КВТ во многом определяется цветом и световой обстановкой.

Потолок, стены выше панелей (если они не облицованы звукопоглощающим материалом) окрашиваются белой водоэмульсионной краской (коэффициент отражения не менее 70 %). Допускается окраска стен цветом панелей до потолка. Для окраски панелей и стен рекомендуется применять светлые тона красок (коэффициент отражения 60 %). Предпочтение следует отдавать холодным тонам: светло-голубому, светло-зеленому, светло-серому. Допускается окраска стен светло-желтым цветом, цветом слоновой кости и т. п.

Оконные переплеты рам, подоконники следует окрашивать белой масляной краской (коэффициент отражения 65—70 %). На окнах монтируются занавеси, гармонирующие по цвету с окраской стен, которые могут полностью закрывать оконные проемы.

Поверхность рабочего стола должна быть матовой цвета натуральной древесины, светло-зеленого, светло-голубого или светлого серого цвета, коэффициент отражения которых должен быть не менее 45 %.

5.6. Освещенность на рабочих столах учащихся и классной доске должна быть порядка 500 лк (разряд П, подразряд «в» по СНиПу П-4-79 «Естественное и искусственное освещение. Нормы проектирования»). Для освещения КВТ следует применять потолочные и подвесные люминесцентные светильники, предназначенные для освещения общественных зданий (специальных светильников для работы на дисплеях наша промышленность не выпускает). Предпочтение следует отдавать светильникам с экранами решетками (металлическими или пластмассовыми).

В целях ограничения отражений блескости от рабочих поверхностей (экран ЭЛТ, клавиатура, поверхность стола) следует в каждом конкретном случае организации КВТ провести светотехнический расчет необходимого количества применяемых светильников, их расположения с учетом объемно-планировочной структуры КВТ, расстановки микро-ЭВМ и организации рабочего места. По этой причине проекция светящихся частей светильников не должна приходиться на экран ЭВМ. Например, при периметральном и двухрядном центральном расположении рабочих мест учащихся с экранами ЭВМ, обращенными в противоположные стороны, проекция светильников должна находиться сзади работающего на ЭВМ.

При монтаже общего освещения необходимо предусмотреть раздельное включение источников света по рядам рабочих мест и для классной доски.

Применять местное освещение при работе учащихся с ЭВМ не рекомендуется!

Для освещения классной доски могут использоваться светильники серии ЛПО12 (там, где они имеются), а во вновь оборудованных кабинетах светильник ЛПО30 (125 модификация).

В целях снижения концентрации пыли от мела рекомендуется применять кодоскопы и другие проекционные аппараты.

В качестве источников люминесцентного освещения должны применяться лампы типа ЛБ, ЛХБ, ЛЕЦ.

Для освещения КВТ запрещается применять лампы типа ЛД!

5.7. Учебные занятия в КВТ при работе на ПЭВМ должны проводиться:

при люминесцентном освещении и зашторенных окнах, когда ЭВМ на рабочих столах располагаются по периметру кабинета или при расположении в два ряда по центру помещения с экранами, обращенными в противоположные стороны, а также при 1—3-рядной расстановке рабочих мест с ЭВМ, когда на улице темно;

при совмещенном освещении (естественное+искусственное) только при 1—3-рядном расположении рабочих мест, когда экран ЭВМ и поверхность рабочего стола находятся перпендикулярно светонесущей стене;

при естественном освещении, когда рабочие места располагаются в один ряд по длине учебного помещения на расстоянии 0,5 и 1,0 м от стены с оконными проемами в зависимости от климатических зон при достаточной освещенности (не ниже 500 лк) и расположении тетради на поверхности стола перед клавиатурой.

При работе на ЭВМ световой поток естественного освещения должен быть слева. Не допускается направление основного светового потока на рабочее место справа, спереди, сзади работающего на ЭВМ. Солнечные лучи и блики не должны попадать в поле зрения учащихся, включая рабочие поверхности (стол, клавиатура, экран ЭВМ, классная доска или экран для проекционных аппаратов).

5.8. Расстановка рабочих мест учащихся в КВТ должна обеспечивать соблюдение правильной рабочей позы учащихся, соблюдение оптимальных расстояний экрана ЭВМ до глаз учащихся и обеспечение подхода педагога во время урока к каждому рабочему месту учащихся.

При 1—3-рядной расстановке рабочих мест в КВТ расстояние между рабочими



столами по длине кабинета должно быть не менее 1,0 м от стены с оконными проемами — 0,5—1,0 м (см. п. 5.7), от стены, противоположной оконным проемам, — 0,5 м, между рядами столов — 0,6 м. Расстояние от передних рабочих мест до демонстрационного стола учителя должно быть не менее 0,8 м, а до классной доски — 2,5 м.

При устройстве входа в КВТ со стороны последних рабочих мест до стены, противоположной классной доске, расстояние должно быть не менее 1,2 м.

КВТ должен быть обеспечен стульями с полумягкими сиденьями и спинкой, меняющимися по высоте для обеспечения правильной посадки учащихся. Под правильной посадкой следует понимать такую, когда учащийся сидит прямо, не сутулясь, с небольшим наклоном до 5—7° вперед, предплечья по отношению к плечу и голени по отношению к бедру находятся под прямым углом, линия зрения направлена перпендикулярно в центр экрана ЭВМ. Допускается ее отклонение в вертикальной плоскости до  $\pm 10^\circ$ . Оптимальный угол от центральной оси экрана в горизонтальной плоскости при рассмотрении информации должен быть в пределах  $\pm 15^\circ$ — $20^\circ$ ; при рассмотрении информации, находящейся в крайних положениях экрана ЭВМ, угол рассматривания не должен превышать  $45^\circ$ .

В целях снижения утомления зрительного анализатора при работе на ЭВМ необходимо соблюдать оптимальное расстояние глаз до экрана ЭВМ, которое составляет 60—70 см, допустимое расстояние — 50 см.

Рассматривать информацию на экране на расстоянии менее 50 см не рекомендуется!

Учащиеся с нарушением зрения должны работать на ЭВМ в очках.

5.9. Длительность работы учащихся на ЭВМ в течение урока не должна быть более 20—25 минут. Целесообразно для работы учащихся на ЭВМ отводить время во второй половине урока.

## 6. Организация работы в КВТ

6.1. Работу КВТ возглавляет заведующий из преподавателей ОИВТ, который является организатором оборудования кабинета, работы учителей и учащихся по ОИВТ и применению ВТ в преподавании отдельных тем программ по основам наук. Под его руководством составляется перспективный план обучения учащихся ОИВТ и план оборудования кабинета; планы на год, пятилетку и перспективу; перечни работ по самооборудованию; распределяется работа между преподавателями и учащимися. Планы утверждаются директором школы.

Заведующий кабинетом несет ответственность за сохранность оборудования, ведение журнала инвентаризационной записи, содержание оборудования в постоянной готовности к применению, своевременность и тщательность профилактического технического обслуживания вычислительной техники, правильное использование ее, регистрацию отказов машин и организацию их отладки или ремонта, за исправность противопожарных средств и средств первой помощи при несчастных случаях, за своевременное проведение вводного и периодического инструктажей учащихся по технике безопасности, регистрацию в журнале времени начала и окончания каждого занятия, включение и выключение электропитания.

6.2. Подключение электропитания к рабочим местам учащихся и выключение его производит преподаватель и отмечает это в журнале использования КВТ на каждом занятии.

6.3. Монтаж, ввод в эксплуатацию, техническое обслуживание и ремонт комплектов учебной вычислительной техники кабинета в гарантийный и послегарантийный сроки осуществляют сервисные предприятия Государственного комитета СССР по вычислительной технике и информатике.

Работы по вводу в эксплуатацию и разовым ремонтам КУВТ выполняются на основе письма-заявки учебного заведения, гарантирующего оплату работ в соответствии с утвержденным прейскурантом либо на основе договоров, заключенных пользователем ЭВМ (учебным заведением или базовым предприятием) с сервисным предприятием, обслуживающим данный регион.

Техническое обслуживание КУВТ в гарантийный и послегарантийный сроки проводятся только на основе договора. Отдельные операции технического обслуживания, по согласованию сторон, допускается проводить преподавателям, заведующим кабинетами, имеющим соответствующее удостоверение.

В течение гарантийного срока эксплуатации устранение дефектов КУВТ проводится за счет предприятия-изготовителя комплекта при условии соблюдения пользователем правил эксплуатации и хранения КУВТ.

Средние учебные заведения (базовые предприятия) обеспечивают строительную готовность кабинетов вычислительной техники под установку и монтаж КУВТ в соответствии с нормами и правилами, включая работы по пожарно-охранной сигнализации, прокладке силовых кабелей до щита в кабине вычислительной техники, окраске стен и потолков, подготовке полов с укладкой металлических защитных труб или металло-

рукавов, по организации заземления.

Сервисные предприятия обеспечивают электромонтажные работы в кабинетах вычислительной техники в соответствии с типовым проектом при условии комплексной поставки КУВТ, включая электрооборудование (силовые щиты, электрические розетки и привод).

Ремонт и техническое обслуживание бытовых телевизоров и магнитофонов, используемых в кабинете, могут осуществлять предприятия бытового обслуживания.

6.4. К оборудованию кабинетов привлекаются базовые предприятия (изготовление мебели, учебного оборудования и т. д.).

При кабинете создается актив учащихся (учеников-лаборантов), который работает по оборудованию кабинета, принимает активное участие в кружковой и факультативной работе.

6.5. Загрузка кабинета учебными, классными, внеклассными и факультативными занятиями, проводимыми учителями и специалистами, приглашенными для преподавания ОИВТ по штатному совместительству, определяется директором школы с участием заведующего кабинетом и преподавателей. Внеклассные занятия по ОИВТ проводятся в присутствии преподавателей.

6.6. В начале изучения курса ОИВТ учащиеся:

распределяются учителем и закрепляются по рабочим местам КВТ с учетом роста, состояния зрения и слуха;

знакомятся под руководством учителя с правилами техники безопасности и работы в кабинете;

сдают зачет по технике безопасности и правилам работы в кабинете, что отмечается в «Журнале регистрации вводного и периодического инструктажей по технике безопасности», в котором указывается дата инструктажей и зачетов, фамилии и инициалы преподавателей, проводивших инструктаж и принявших зачет, фамилии и инициалы учеников, сдавших зачет, содержание инструктажа. Эти данные скрепляются подписью учителя (Приложение 5, п. 2).

Ученики несут полную ответственность за состояние рабочего места и размещенного на нем оборудования.

6.7. Преподаватели строго следят за выполнением учащимися требований техники безопасности и правил работы в КВТ.

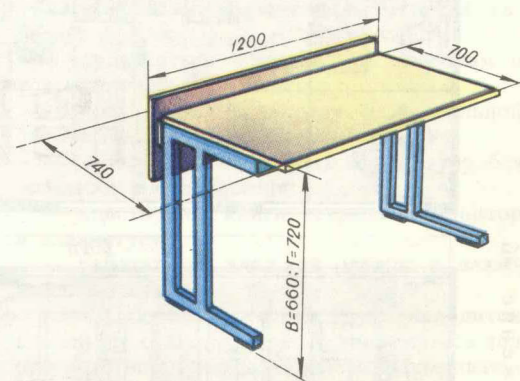
6.8. На каждом занятии в журналах использования микро-ЭВМ отмечается время начала и окончания работы, содержание ее, состояние рабочего места, отказы машины.

## Приложение 1

Перечни технических средств, учебно-наглядных пособий и мебели для кабинетов вычислительной техники всех типов учебных заведений.

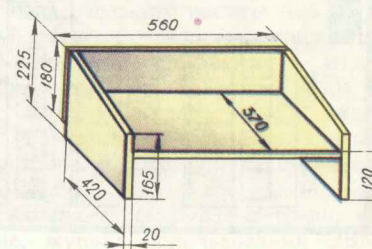
Перечни были опубликованы в ИНФО № 1.

## Приложение 2.



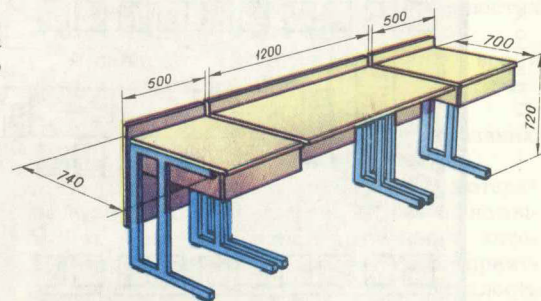
Рабочее место учащегося

## Приложение 2а



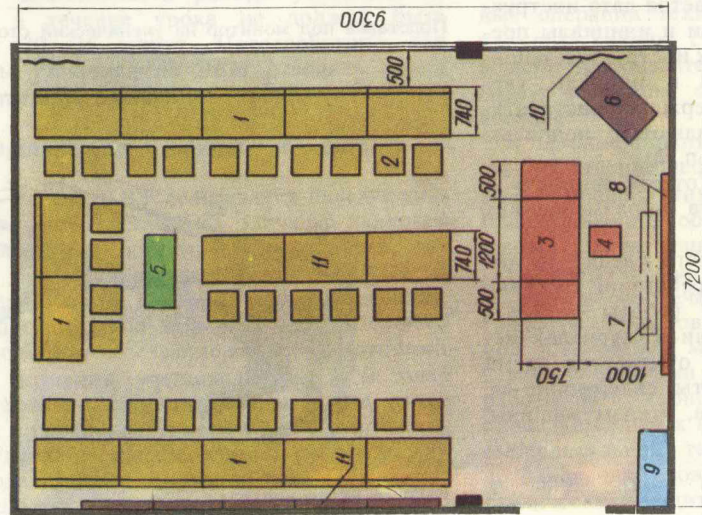
Подставка под монитор на ученический стол

## Приложение 3



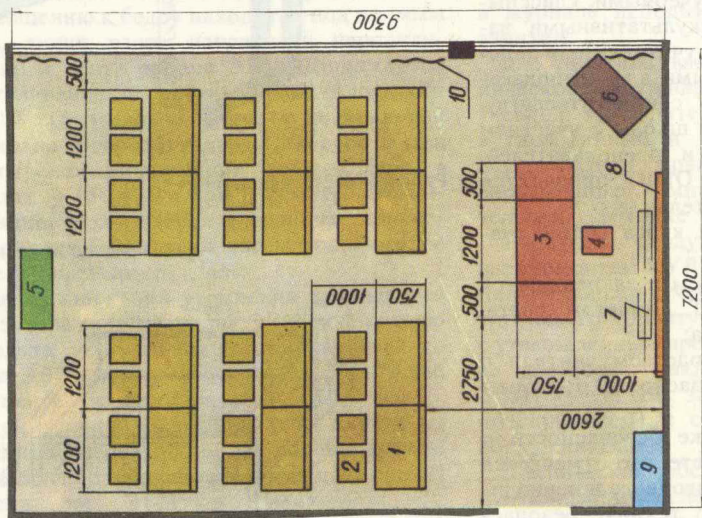
Рабочее место учителя





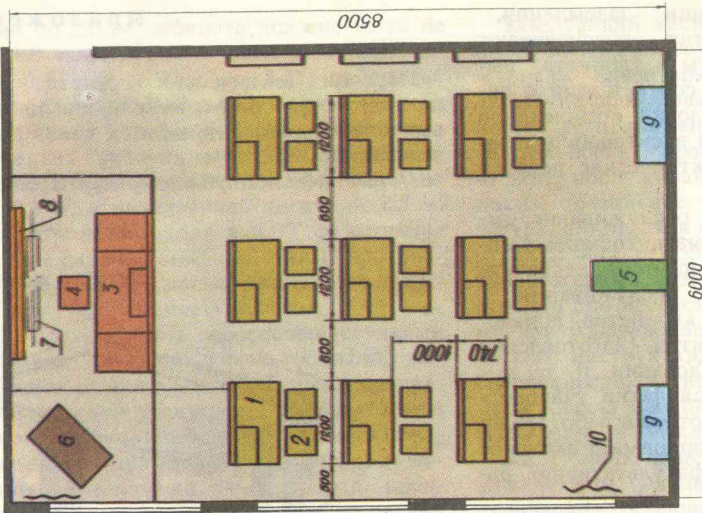
Кабинет вычислительной техники  
на 30 рабочих мест и 12 ЭВМ

- 1 — столы с ученическими ЭВМ; 1.1 — столы ученические; 2 — стулья ученические; 3 — стол с учительской ЭВМ, принтером и графопроектором; 4 — стул учителя; 5 — подставка для проектора; 6 — телевизор демонстрационный; 7 — экран; 8 — доска классная; 9 — шкаф для учебных пособий; 10 — шторы затемнения; 11 — шиты экспозиционные



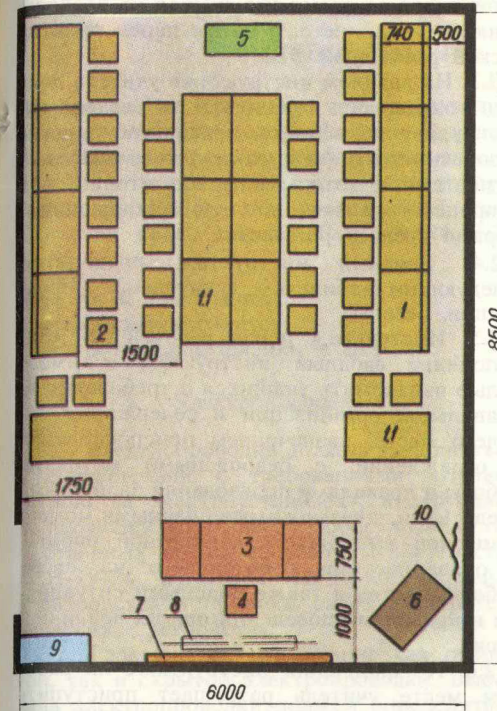
Межшкольный кабинет вычислительной техники  
на 24 рабочих места и 12 ЭВМ

- 1 — столы с ученическими ЭВМ; 2 — стулья ученические; 3 — стол с учительской ЭВМ, принтером и графопроектором; 4 — стул учителя; 5 — подставка для проектора; 6 — телевизор демонстрационный; 7 — экран; 8 — доска классная; 9 — шкаф для учебных пособий; 10 — шторы затемнения



Кабинет вычислительной техники  
школы № 226 Москвы

- 1 — столы с ученическими ЭВМ; 2 — стулья ученические; 3 — стол с учительской ЭВМ; 4 — шкаф для учебных пособий; 5 — подставка для проекторов; 6 — телевизор демонстрационный; 7 — экран; 8 — доска классная; 9 — шкаф для учебных пособий; 10 — шторы затемнения



Кабинет вычислительной техники  
на 32 рабочих места и 9 ЭВМ

- 1 — столы с ученическими ЭВМ; 1.1 — столы ученические; 2 — стулья ученические; 3 — стол с учительской ЭВМ, принтером и графопроектором; 4 — стул учителя; 5 — подставка для проектора; 6 — телевизор демонстрационный; 7 — экран; 8 — доска классная; 9 — шкаф для учебных пособий; 10 — шторы затемнения

## Приложение 5

### 1. Правила работы учащихся в кабинете вычислительной техники<sup>1</sup>

1.1. В кабинете вычислительной техники (КВТ) установлена дорогостоящая, сложная и требующая осторожного и аккуратного обращения аппаратура — компьютеры (ЭВМ), принтер, другие технические средства.

Поэтому

- бережно обращайтесь с этой техникой;
- спокойно, не торопясь, не толкаясь, не задевая столы, входите в кабинет и занимайте отведенное вам место, ничего не трогая на столах.

<sup>1</sup> Этот раздел рекомендуется размножить (15—20 экз.), чтобы передавать на рабочие места учащихся при изучении правил.

1.2. На вашем рабочем месте размещены составные части ЭВМ — системный блок, клавиатура и монитор (дисплей).

Во время работы лучевая трубка монитора (дисплея) работает под высоким напряжением.

Неправильное обращение с аппаратурой, кабелями и мониторами может привести к тяжелым поражениям электрическим током, вызвать загорание аппаратуры.

Поэтому строго запрещается

- трогать разъемы соединительных кабелей;
- прикасаться к питающим проводам и устройствам заземления;
- прикасаться к экрану и к тыльной стороне монитора, клавиатуры;
- включать и отключать аппаратуру без указания преподавателя;
- класть диск, книги, тетради на монитор и клавиатуру;
- работать во влажной одежде и влажными руками.

1.3. Хорошо запомните, где находится в кабинете огнетушитель. Познакомьтесь под руководством преподавателя, как пользоваться им. При появлении запаха гари немедленно прекратите работу, выключите аппаратуру и сообщите об этом преподавателю; в случае необходимости окажите ему помощь в тушении огня.

1.4. Перед началом работы

- убедитесь в отсутствии видимых причин повреждений рабочего места;
- сядьте так, чтобы линия зрения приходилась в центр экрана, чтобы не наклоняясь пользоваться клавиатурой и воспринимать передаваемую на экран монитора информацию;
- разместите на столе тетрадь, учебное пособие, журнал использования ЭВМ так, чтобы они не мешали работе на ЭВМ;
- внимательно слушайте объяснения учителя и старайтесь понять цель и последовательность действий; в случае необходимости обращайтесь к преподавателю;

— хорошо разберитесь в особенностях применяемых в работе аппаратов;

— запишите в журнал регистрации время начала работы на ЭВМ;

— расчехлите машину;

— начинайте работу только по указанию преподавателя «приступить к работе».

1.5. Во время работы ЭВМ (компьютера) лучевая трубка монитора является источником электромагнитного излучения, которое при работе вблизи экрана неблагоприятно действует на зрение, вызывает усталость и снижение работоспособности.

Поэтому надо работать

- на расстоянии 60—70 см, допустимо



не менее 50 см, соблюдая правильную посадку, не сутулясь, не наклоняясь; учащимся, имеющим очки для постоянного ношения, — в очках.

1.6. Работа на ЭВМ требует большого внимания, четких действий и самоконтроля.

Поэтому нельзя работать

- при недостаточном освещении;
- при плохом самочувствии (в этом случае надо обратиться к врачу).

1.7. Во время работы

- строго выполняйте все указанные выше правила, а также текущие указания учителя;
- следите за исправностью аппаратуры и немедленно прекращайте работу при появлении необычного звука или самопроизвольного отключения аппаратуры. Немедленно докладывайте об этом преподавателю;

— плавно нажимайте на клавиши, не допуская резких ударов;

— не пользуйтесь клавиатурой, если не подключено напряжение;

— работайте на клавиатуре чистыми руками;

— никогда не пытайтесь самостоятельно устранять неисправности в работе аппаратуры;

— не вставайте со своих мест, когда в кабинет входят посетители.

1.8. По окончании работы

- отключите тумблер «Сеть»;
- запишите в журнале регистрации использования ЭВМ время окончания работы;
- протрите аппаратуру куском мягкой чистой ткани;
- зачехлите аппаратуру.

1.9. Вы должны хорошо знать и грамотно выполнять эти правила, точно следовать указаниям преподавателя, чтобы

- избежать несчастных случаев;
- успешно овладеть знаниями, умениями, навыками;

— сберечь государственное имущество — вычислительную технику и оборудование.

Вы отвечаете за состояние рабочего места и сохранность размещенного на нем оборудования.

Невыполнение правил — грубейшее нарушение порядка и дисциплины.

## 2. Указания по проведению инструктажа учащихся по технике безопасности

2.1. Для воспитания у учащихся сознательного отношения и усвоения правильных и безопасных методов и приемов работы учителя обязаны проводить инструктирование и обучение учащихся по соблюдению требований техники безопасности (Приложение 6) и гигиены труда.

2.2. Инструктаж и обучение по технике безопасности и производственной санитарии проводится со всеми учащимися на вводном занятии в кабинете, а затем перед практической работой на ЭВМ.

2.3. На вводном инструктаже учитель должен ознакомить учащихся с правилами распорядка в кабинете, правилами техники безопасности и гигиены труда; с опасными моментами, с которыми можно столкнуться в процессе работы, и с соответствующими мерами предосторожности.

2.4. Вводный инструктаж проводится заведующим кабинетом (учителем) в виде лекции, беседы.

2.5. Инструктаж перед работой на ЭВМ дополняет вводный инструктаж и имеет целью ознакомить учащихся с требованиями правильной организации и содержания рабочего места, назначением приспособлений и ограждений, с безопасными методами работы и правилами пользования защитными средствами, с возможными опасными моментами при выполнении конкретной работы, с обязанностями работающего на своем рабочем месте, а также опасными ситуациями и правилами поведения при их возникновении.

2.6. По окончании инструктажа на рабочем месте учитель разрешает приступить к самостоятельной работе только после того, как убедится, что все учащиеся усвоили инструктаж.

2.7. Инструктаж на рабочем месте должен быть кратким, содержать четкие и конкретные указания и в необходимых случаях сопровождаться показом правильных и безопасных приемов выполнения работы.

2.8. В процессе выполнения работы учитель и лаборант обязаны систематически контролировать выполнение каждым учеником данных ему при инструктаже указаний о безопасном способе выполнения работы.

### ЖУРНАЛ

регистрации инструктажа по технике безопасности

№ п.п	Фамилия, имя инструктируемого	Дата	Содержание инструктажа с указанием названия инструкции	Фамилия, имя, отчество проводившего инструктаж, его должность	Подпись проводившего инструктаж	Подпись инструктируемого

2.9. Все сведения по проведению инструктажа учащихся заносятся в специальный журнал.

## Приложение 6

Извлечение из Правил по технике электробезопасности при проведении занятий в учебных кабинетах (классах) общеобразовательных школ и практики школьников на промышленных объектах», утвержденных зам. министра просвещения СССР от 26.10.72.

### Часть 2. Требования к устройству кабинетов (электротехники, физики и др.)

#### Раздел 3. Электропроводка

3.1. Электропроводка должна выполняться в соответствии с требованиями «Правил устройства электроустановок», инструкцией по проектированию электрооборудования общественных зданий массового строительства ВСН-19-74 и СНиП III-33-76 «Правила производства и приемки работ электротехнических устройств».

3.2. Разрешается применение как открытой, так и скрытой электропроводки. Выбор вида электропроводки должен определяться архитектурными и строительными особенностями помещения и условиями безопасности ее эксплуатации.

3.3. Провода и кабели, применяемые для электропроводки, должны иметь изоляцию, рассчитанную на напряжение переменного тока не ниже 500 в.

3.4. Все работы по прокладке, креплению, присоединению и ремонту проводов производятся только при снятом напряжении.

3.5. Выбор сечения проводов в зависимости от тока нагрузки и способа прокладки должен производиться в соответствии с таблицами № 1 и 2.

3.6. В местах, где возможно механическое повреждение проводов, кабелей, они должны быть дополнительно защищены.

3.7. Высота прокладки изолированных защищенных проводов от уровня пола не нормируется.

3.8. Расстояние от открыто проложенных на изолирующих опорах незащищенных проводов до поверхности стен и других предметов не должно быть менее 10 мм.

3.9. Изоляционные трубы с металлической оболочкой, защищенные провода, кабели и гибкие металлические рукава электропроводки закрепляются на опорных поверхностях трубы — через 0,8—1,0 м; провода, кабели и гибкие металлические рукава — через 0,5—0,7 м.

Таблица 1

Провода и шнуры с резиновой или полихлорвиниловой изоляцией

Сечение жилы	Токовые нагрузки (А) на провода, проложенные в одной трубе					
	Открыто	2 однопроволочных	3 однопроволочных	4 однопроволочных	1 двупроволочный	1 трехпроводный
С медными жилами						
0,5	11	—	—	—	—	—
0,75	15	—	—	—	—	—
1	17	16	15	14	15	14
1,5	23	19	17	16	18	15
2,5	30	27	25	25	25	21
4,0	41	38	35	30	32	27
6,0	50	46	42	40	40	34
10	80	70	60	50	55	50
16	100	85	80	75	80	70
25	140	115	100	90	100	85
С алюминиевыми жилами						
2,5	24	20	19	19	—	—
4,0	32	28	28	23	—	—
6,0	39	36	32	30	—	—
10	55	50	47	39	—	—
16	80	60	60	55	—	—
25	105	85	80	70	—	—

Таблица 2

Шнуры, провода и переносные шланговые кабели

Сечение жилы мм <sup>2</sup>	Токовые нагрузки		
	однопроволочные	двупроволочные	трехпроводные
0,5	—	12	—
0,75	—	16	14
1,0	—	18	16
1,5	—	23	20
2,5	40	33	28
4,0	50	43	36
6,0	65	55	45
10	90	75	60

3.10. Открыто проложенные трубы закрепляются как на горизонтальных, так и на вертикальных участках на расстоянии 2,5—4 м в зависимости от диаметра трубы.

3.11. При скрытой электропроводке должна быть обеспечена возможность замены проводов.

3.12. Соединение проводов сечением от 1 до 10 квадратных миллиметров разрешается только пайкой с предварительной скруткой. При сечении до одного квадратного миллиметра допускается соединение скруткой.



3.13. Присоединение проводов к коммутационной аппаратуре, электродвигателям, электроприборам и т. п. производится с помощью наконечников для проводов: однопроволочных — сечением свыше 10 квадратных миллиметров, многопроволочных — сечением свыше 2,5 квадратного миллиметра.

#### Раздел 4. Распределительные щиты и пульты питания

4.1. Питание потребителей электроэнергии осуществляется от распределительных щитов и пультов.

4.2. Устройство распределительных щитов и пультов питания производится в строгом соответствии с «Правилами устройства электроустановок».

4.3. Токоведущие части в распределительных щитах и пультах закрываются сплошными ограждениями и запираются на замок.

4.4. Корпуса щитов и пультов выполняются из несгораемых или трудносгораемых материалов.

4.5. Все металлические части щитов и пультов должны быть покрашены или иметь другое антикоррозийное покрытие.

4.6. Панели щитов и пультов должны иметь четкие надписи, указывающие назначение отдельных цепей или панелей. Надписи выполняются на лицевой стороне устройства, а при обслуживании с двух сторон — также и на задней стороне устройства.

4.10. Щиты и пульты надежно укрепляются.

4.11. Плавкие вставки предохранителей должны точно соответствовать значениям данного присоединения, указанным на схеме, расположенной на внутренней стороне дверки щита.

Применение некалиброванных и завышенных по току плавких вставок во всех типах предохранителей запрещается.

4.12. Все распределительные щиты и пульты питания должны быть снабжены кнопкой аварийного отключения, обеспечивающей отключение электропитания всего кабинета или класса (за исключением общего освещения).

#### Часть 3. Техническое обслуживание и ремонт электрооборудования

##### Раздел 1. Общие положения

1.1. Техническое обслуживание и ремонт электрооборудования учебных кабинетов с точки зрения мер безопасности относится к работам, выполняемым при частичном или полном снятии напряжения.

1.2. К техническому обслуживанию и ре-

монту оборудования допускается только электротехнический персонал (лица, ответственные за электрохозяйство школы, преподаватели электротехники и электромонтеры).

1.3. Все лица, имеющие право технического обслуживания и ремонта электрооборудования, должны изучить настоящие «Правила» и иметь квалификацию в соответствии с выполняемой работой.

#### Раздел 3. Организация безопасного проведения работ

3.1. Работы по техническому обслуживанию и ремонту кабинетов должны выполняться по распоряжению с записью в журнале производства работ. В журнале должно быть указано, кем отдано распоряжение, содержание и место работы, категория производства работ в отношении мер безопас-

Таблица 3  
Содержание аптечки первой помощи

Наименование медицинских средств и медикаментов	Назначение	Кол-во
Индивидуальные перевязочные антисептические пакеты	Для наложения повязок	3 шт.
Бинты	то же	3 шт.
Вата в пакетах	то же	3 шт.
Жгут	Для остановки кровотечения	1 шт.
Настойка йода	Для смазывания окружности раны, свежих ссадин, царапин на коже и т. д.	1 склянка или 10 ампул
Нашатырный спирт	Применять при обмороках, накатав на ватку 2—3 капли и поднося к носу пострадавшего	1 флакон или 10 ампул
Раствор 2—4 % борной кислоты	Для промывания глаз, для примочек на глаза, при ожогах глаза вольтовой дугой, для полоскания рта при ожогах щелочью	1 флакон 350 мл
Вазелин	Для смазывания кожи при ожогах 1-й степени, раздражениях, ссадинах	1 тюбик
Валидол	Применять при сильных болях в области сердца по одной таблетке под язык до полного рассасывания	1 упаковка

ности, перечень технических и организационных мероприятий, время выполнения работы, фамилия, инициалы, квалификационная группа производителя работ (наблюдающего) и членов бригады. После окончания работ в журнале должна быть сделана соответствующая отметка с указанием даты и времени их окончания.

3.2. При проведении работ по техническому обслуживанию и ремонту электрооборудования в помещении должно находиться не менее двух лиц, одно из которых — производитель работ, а другое — член бригады. В случае необходимости одним из этих лиц покинуть помещение, даже кратковременно, второе лицо обязано прекратить работы по электрооборудованию и выйти из помещения.

#### Раздел 6. Периодичность и объем профилактических работ

6.1. Не реже 1 раза в 3 месяца необходимо проделать следующий комплекс работ:

а) Стереть пыль со всех поверхностей электрооборудования (щитов, пультов питания, рубильников, пускателей и т. д.).

б) Внимательно осмотреть все оборудование с целью выявления механических повреждений и устранить их в случае необходимости.

в) Проверить затяжку винтов (болтов) электрических соединений и при необходимости произвести их подтяжку; подгоревшие контактные соединения разъединить, зачистить мелкой наждачной шкуркой, смазать и соединить вновь.

г) Не реже 1 раза в 3 года необходимо проверить мегаомметром сопротивление изоляции всех единиц электрооборудования.

#### Меры первой помощи при поражении электротоком

При поражении электротоком следует быстро отключить ток и освободить пострадавшего от проводов.

В случае остановки дыхания следует срочно вызвать врача (медицинскую сестру) и немедленно приступить к выполнению искусственного дыхания. Искусственное дыхание должно выполняться ритмично (16—18 раз в мин) и длительно (до прихода врача или восстановления дыхания).

### Передачи Главной редакции научно-популярных и учебных программ Центрального телевидения на 1987 год. Курс «Основы информатики и вычислительной техники»

#### IX класс

Последовательное построение алгоритма. Вспомогательные алгоритмы	17.III
Этапы решения задач на ЭВМ	21.IV
Построение алгоритмов для решения задач из курса математики	28.IV
Построение алгоритмов для решения задач из курса физики	12.V

#### X класс

Алгоритмы вычисления функций	21.I
Алгоритмы работы с текстами	4.II
Язык программирования Рапира (вне урока)	25.II
Язык программирования Бейсик	4.III
Краткая история вычислительной техники	15.IV

Научный консультант — академик А. П. ЕРШОВ



Я. КОБРИНСКИЙ, А. КУЗНЕЦОВ  
НИИ СИМО АПН СССР

## Особенности пакетов прикладных программ

18

В «Основных направлениях реформы общеобразовательной и профессиональной школы» определены два направления внедрения ЭВМ в среднюю школу:

введение основ информатики и вычислительной техники в содержание обучения в школе как компонента общеобразовательной подготовки учащихся;

использование ЭВМ в школе как современного средства обучения.

Реализация обоих указанных направлений требует создания соответствующего программного обеспечения — прикладных обучающих программ. Состав и назначение пакетов прикладных программ (ППП) является одним из основных вопросов компьютеризации школы.

Чрезвычайная актуальность создания программного обеспечения учебного процесса в школе инициировала разработки разного рода программ, которые ведутся десятками различных исследователей и научных организаций, с различными подходами не только к созданию и типологии обучающих программ, но и к средствам программной (инструментальной) их реализации. В результате может сложиться ситуация, когда созданное программное обеспечение будет бессистемным, часто лишенным взаимосовместимости.

Данная статья содержит эскиз требований, которым должны удовлетворять современные пакеты прикладных программ. Учет этих требований облегчит процесс подключения вновь создаваемых программ к разработанному программному обеспечению и будет способствовать исключению ненужного параллелизма в работе.

### Исходные предпосылки, определяющие назначение и состав пакетов прикладных программ

Цель использования PPP — повышение эффективности и качества обучения.

При решении вопроса о составе и назначении PPP необходимо иметь в виду две функции ЭВМ как средства обучения:

мощный современный инструмент познания;

средство автоматизации процесса обучения.

Первая из них обеспечивает формирование у учащихся умений и навыков применения ЭВМ для повышения эффективности деятельности человека в различных областях, овладение школьниками современными методами научного познания мира, связанными с применением компьютеров. Повышение эффективности учебного процесса и качества обучения при использовании PPP, реализующих первую — «инструментальную» — функцию применения ЭВМ в обучении, доказано имеющимся отечественным и зарубежным опытом, поэтому разработка таких пакетов — первоочередная задача.

Вторая функция — автоматизация обучения — непосредственно связана с развитием идей программированного обучения.

Несомненно, что применение современных компьютеров в качестве средства программированного обучения открывает в ряде случаев принципиально новые возможности, позволяющие значительно усилить известные положительные аспекты данного вида обучения. Однако позволит ли применение ЭВМ устранить те его негативные аспекты, кото-

рые не позволили программированному обучению широко войти в практику школы? Этот вопрос требует тщательного исследования и, главное, серьезной экспериментальной проверки. Поэтому при создании описываемой проблемно-ориентированной системы ставилась задача обеспечения экспериментальной проверки возможности автоматизации процесса обучения.

Итак, при разработке системы необходимо учитывать следующие основные положения.

Во-первых, необходимо стремиться к возможно более полному использованию ЭВМ, имея в виду различные аспекты ее применения: для хранения программных моделей и диалоговых программ, предназначенных для обучения различным предметам; для обучения программированию, для контроля знаний; для сбора и обработки учебной информации, необходимой для анализа эффективности учебного процесса. Во-вторых, программное обеспечение должно давать возможность использовать современные достижения в области методики обучения, которые должны найти отражение в программированных курсах обучения, моделях для активизации усвоения нового материала и для использования в качестве средства решения задач. В-третьих, программное обеспечение должно учитывать условия применения ВТ в общеобразовательной школе: наличие локальной сети, ограничения по предоставляемым ресурсам центральной и периферийных машин, разнообразные возможности, которые обеспечивает микро-ЭВМ при обработке текстовой, графической и звуковой информации.

Учет этих соображений позволяет сделать два вывода.

Сведение разработки программного обеспечения к созданию и распространению отдельных обучающих программ по различным школьным предметам не способствует эффективно использованию ЭВМ в обучении. Совокупность прикладных программ должна быть организована по типу проблемно-ориентированной системы (ПОС). Этим термином называют такие программные средства, которые используются для решения задач определенного класса (см. статью А. П. Ершова и В. П. Ильина «Пакеты прикладных программ как методология решения прикладных задач» в сборнике «Пакеты прикладных программ: проблемы и перспективы». М.: Наука, 1982). Отличие ПОС от библиотеки программ в том, что она содержит три составные части: функциональное наполнение (совокупность программно-реализованных моделей, отражающих особенности предметной области), системное наполнение (сервисные средства, предоставляемые пользователям и инва-

риантные относительно конкретной предметной области) и язык запросов (директив), упрощающий процесс общения с ПОС. Программы системного наполнения и транслятор с языка запросов должны поставляться централизованно, в комплекте программного обеспечения школьного КВТ; в них должны быть изложены требования и условия подключения программ функционального наполнения. Последние могут разрабатываться «на местах», т. е. в вычислительных центрах и учреждениях, непосредственно занимающихся школьным образованием. Такая организация работы позволит значительно сэкономить средства на создание программного обеспечения и, самое главное, обеспечит эффективное использование «функциональных» программ в учебном процессе.

Применение ЭВМ в учебном процессе позволяет реализовать автоматизацию обучения на качественно новом уровне. Однако методическая эффективность программированных курсов, реализуемых на ЭВМ, требует тщательной экспериментальной проверки и оценки. В настоящее время за рубежом разработаны мощные инструментальные средства создания программированных курсов обучения, такие, как «Interactive Instruction System», «Private Tutor» и др. Они позволяют специалистам, разрабатывающим методические материалы (учебники, пособия, факультативы и т. д.), создавать программированные курсы, структурированные по урокам. Методика написания сценариев уроков должна быть предметом отдельного исследования, однако она, несомненно, должна опираться на достижения отечественной школы программированного обучения (П. Я. Гальперин и др.).

В составе сценария урока должны быть три части: а) демонстрационная модель, которая служит для повышения эффективности и качества усвоения нового материала и позволяет учащемуся в процессе работы с ней активно познавать тот или иной процесс или явление, работать в диалоговом режиме, осуществлять ввод данных и получение результатов в наглядной форме; б) сценарий опроса, содержащий материал, необходимый для усвоения новых понятий и заставляющий учащегося пользоваться ими (это могут быть контрольные вопросы, тексты незавершенных ответов или диктантов — с проблемами или с заведомо введенными ошибками); в) тексты самостоятельных заданий с возможностью введения вариативности исходных данных. При написании сценариев уроков необходимо учитывать ограничения, которые накладывает ЭВМ (например, вопросы должны быть сформулированы так, чтобы они могли быть проанализированы

19



машиной). Разработанный в соответствии с инструкцией программный курс обучения может распространяться на магнитном носителе, прикладываемом к тексту учебника.

Подводя итоги, можно сказать, что прикладное программное обеспечение, разрабатываемое по типу ПОС, должно включать программы системного наполнения, поставляемые в комплекте программного обеспечения КВТ, и программы функционального наполнения, частично поставляемые с системными программами, дополняемые на местах их использования.

Для эксперимента по проверке эффективности автоматизации обучения ПОС может включать в себя программный курс обучения (структурированный по урокам), поставляемый на магнитных носителях в виде приложения к учебному пособию.

## 20 Возможности, предоставляемые прикладным программным обеспечением

Прикладные программы, входящие в состав ПОС «Использование ЭВМ в школьном образовании», должны предоставлять следующие возможности:

а) обеспечение возможности решения учащимися учебных задач с использованием ЭВМ (предполагается, что соответствующее программное обеспечение позволит работать с ЭВМ как на уровне непрофессионального пользователя (без написания самостоятельных программ), так и на уровне прикладного программиста с возможностью использования готовых к выполнению программ);

б) обеспечение частичной автоматизации проверки правильности выполнения заданий путем прогона эталонной и учебной программ на одних и тех же исходных данных;

в) введение в учебный процесс предварительно разработанного программного курса обучения, структурированного по урокам (сценарий урока содержит опрос по старому материалу, демонстрационные модели и текст самостоятельного задания);

г) учет успеваемости по результатам опроса и ведение учета результатов выполнения заданий.

Система в процессе функционирования использует следующие архивы, хранимые на магнитных носителях:

- архив сценариев опроса;
- архив демонстрационных моделей;
- архив текстов заданий для учащихся;
- архив данных для контроля правильности выполненных заданий;
- архив оперативного учета успеваемости по результатам опроса;
- архив сведений о посещаемости занятий

и результатах выполнения самостоятельных заданий.

Первые четыре архива составляют информационное обеспечение программного курса обучения по данному предмету. Сценарий опроса, тексты заданий для учащихся и данные для контроля выполнения заданий поставляются в виде приложения к учебному пособию на магнитном носителе. Архив демонстрационных моделей формируется на основе программ функционального наполнения. Информация в последние два архива заносится во время учебного процесса и служит для оценки успеваемости и уровня подготовки учащихся.

В заключение этого раздела отметим, что ПОС «Использование ЭВМ в школьном образовании» должна быть системой «открытого» типа, т. е. допускать наращивание функций и включение новых программ в состав системного и функционального наполнения.

## Реализация учебного процесса в условиях использования прикладного программного обеспечения

Еще раз подчеркнем, что прикладные программы, составляющие функциональное наполнение ПОС, делятся на:

«инструментальные», которые будут указаны в учебной программе по каждому учебному предмету и обязательно будут использоваться школьниками при его изучении;

программы автоматизации обучения, которые на первом этапе будут разрабатываться исключительно в экспериментальных целях, а затем (в случае их успешной апробации) могут быть только рекомендованы учителю, который сможет применять или не применять их в учебном процессе, в зависимости от выбранной им методики обучения.

Перейдем к описанию схемы учебного процесса в случае применения программ автоматизации обучения.

1. Учитель в порядке подготовки к уроку формирует сценарий урока, выбирая соответствующие разделы из архивов сценария опроса, текстов заданий, данных для контроля и демонстрационных моделей. Пользуясь программой-вариатором, он заполняет соответствующие поля исходными данными. Полученный файл сегментируется по кадрам, каждый из которых должен умещаться на экране монитора, и записывается в библиотеку учащихся.

2. Урок начинается с «переключки»: каждый учащийся на своем рабочем месте набирает свой код и с учительской ЭВМ производится опрос ученических ЭВМ, регистрация посещения занятия в «машинном журнале». Далее учащийся использует свою библиоте-

ку на внешнем носителе, где содержатся необходимые системные средства. Пользуясь стандартной диалоговой программой, он подвергается опросу. При необходимости освободить экран для выполнения вычислений, графических построений и т. д. он пользуется соответствующими функциональными клавишами, запуская сервисные программы; с их помощью можно вернуться к текущему кадру сценария урока, пройти по сценарию на любое число кадров вперед, назад и т. д. Диалоговая программа проведения опроса проверяет правильность действий учащегося и результаты заносит в рабочую область, откуда они по окончании урока собираются в архив оперативного учета результатов опроса.

Демонстрационная модель является программой. Под ее управлением учащийся изучает соответствующие явления, процессы, знакомится с примерами использования ключевых понятий и типичными ошибками. В конце урока он получает и записывает в свою личную библиотеку самостоятельное задание (если это предусмотрено сценарием). Шифр задания запоминается в рабочем поле для пересылки в архив учета выполнения заданий.

3. Во время практикума учащийся самостоятельно решает задачи. При необходимости использования готовых программ или моделей он может переписать в свою библиотеку соответствующие программы функционального наполнения. Как правило, эти модули указываются в тексте самого задания.

4. Учитель после урока, пользуясь архивом данных для контроля правильности выполненных заданий, осуществляет контроль (если это возможно) путем прогона программ ученика и эталонной и сверки результатов.

## Программы системного наполнения

Как было указано ранее, программы системного наполнения содержат сервисные средства, независимые от конкретной предметной области. По назначению их можно разделить на:

средства диалогового взаимодействия с ЭВМ на уровне непрофессионального пользователя, которые позволяют как учителю, так и учащемуся реализовать возможности, о которых говорилось в предыдущем разделе, не прибегая к составлению собственных программ;

средства программирования, предоставляющие пользователю дополнительные по сравнению с базовой системой программирования возможности для разработки собственных программ.

Средства диалогового взаимодействия де-

лятся на общие, средства для учителя, средства для учащихся.

В общие средства диалогового взаимодействия входят следующие программы.

**Монитор**, интерпретирующий запросы ученика или учителя в ходе урока, при выполнении самостоятельного задания учеником или при подготовке к уроку учителя и запускающий соответствующие программы для выполнения запросов.

**Графический редактор**, осуществляющий отображение на экране и корректировку графической информации без составления программы пользователем. Средством ввода информации является световое перо или (при его отсутствии) светящаяся точка в графике заданного разрешения. Координаты светящейся точки («графическая курсора») программно фиксируются, и средствами клавиатуры она перемещается по экрану. Таким образом можно рисовать и стирать графические элементы (отрезок прямой, дугу окружности и т. д.). Другое средство, предоставляемое графическим редактором — меню: набор символов по периферии экрана, каждый из которых обозначает графический элемент или функцию (например, окружность, штриховка, пунктирная линия, окрашивание замкнутой области в определенный цвет и т. д.). Фиксация той или иной функции осуществляется путем перемещения графического курсора в область экрана, занимаемую заданным символом. Эти же средства можно использовать для выдачи текста на экран в графике высокого разрешения. Индикатором положения отдельных символов является графический курсор, а для передачи изображений букв или цифр могут служить программируемые графические элементы (спрайты), выбираемые нажатием клавиш. Графический редактор осуществляет также сохранение содержимого экрана во внешней памяти и в случае необходимости его восстановления.

**Информационно-справочная система**, осуществляющая выдачу пользователю необходимой информации по запросу. Она должна иметь две подсистемы: поиск по ключу в словнике (гlossарии) для разъяснения определенного понятия и поиск по заранее незапланированному запросу, организованный по типу экспертной системы. В последнем случае инициативу по определению того, что пользователь хочет выяснить, берет на себя машина: вопросы задаются ею в зависимости от получаемых ответов. В случае успеха пользователь получает интересующие его сведения. При невозможности ответить на поставленный вопрос система «просит» пополнить запас хранимой ею справочной информации, обеспечивая тем самым наращивание



необходимого для будущего запаса знаний. Таким образом, экспертные системы перед началом их практического использования требуют «обучения» для занесения необходимого объема первоначальной информации. Средства диалогового взаимодействия для учителя включают следующие компоненты.

*Программа формирования сценария урока* осуществляет выборку файлов, отражающих содержание отдельных фрагментов урока (опроса, демонстрационной модели и текста задания) и сегментирование сценария урока в соответствии с размерами экрана. Для обеспечения вариативности упражнений, включаемых в сценарий опроса, и самостоятельных заданий в составе программы имеется специальный модуль — вариатор. Текст вопроса или задания, подаваемого на вход вариатора, имеет «пробелы» — последовательности символов, которые заменяются на числовые или текстовые константы, обеспечивая тем самым попарное различие вариантов вопросов и заданий. Файл, получаемый выборкой отдельных фрагментов урока, их слиянием, сегментированием и заполнением «пробелов» с помощью вариатора, является сценарием урока.

*Программа контроля* правильности и оценки качества выполненных заданий осуществляет контроль по конечному результату или пооперационный контроль правильности выполнения самостоятельного задания и оценивает, если это возможно, рациональность (эффективность) способа (алгоритма) решения. Примером может служить контроль правильности выполнения задания по информатике — составления алгоритма и программы. Текст задания содержит имена переменных (аргументов и результатов) и значения аргументов. Указанная информация используется для прогноза эталонной программы и проверяемой программы с последующей сверкой результатов. Пооперационный контроль предполагает, что поставленная задача разбивается на шаги (этапы) и контроль осуществляется после каждого этапа. Естественно, что контролировать можно не только числовые, но и текстовые данные (например, проверять диктанта).

*Программа выдачи учетной информации* по запросу учителя выдает сведения о посещениях занятий, оценки знаний учащихся по результатам опроса и выполнения самостоятельных заданий в разрезе отдельных предметов или групп учащихся и при необходимости вычисляет обобщенные показатели (количество отстающих и успевающих за данный период по выбранному предмету, процент учащихся, успешно справившихся с заданиями по данной теме, и т. д.).

Средства диалогового взаимодействия учителя включают следующие компоненты.

*Программа управления диалогом* при проведении опроса использует в качестве исходной информации сценарий опроса, представленный в виде связанного списка. Каждая запись списка имеет поля: номер упражнения (вопроса); содержание упражнения; эталонные ответы или средства для анализа правильности действий учащегося; реакция программы по результатам анализа ответа; номер следующего упражнения по результатам анализа ответа. По окончании диалога программа должна выставить оценку — суммарную или по каждому упражнению. Отдельные упражнения, составляющие содержание опроса, могут иметь разную форму: как контрольных и наводящих вопросов, так и диктанта (текст с пробелами или преднамеренно введенными ошибками). Поэтому программа управления диалогом должна иметь набор модулей для анализа ответа учащегося. Оценка правильности ответа должна производиться с учетом возможных несущественных ошибок (грамматические ошибки и сокращения), конечно, в пределах возможностей ЭВМ. Реакция программы по результатам анализа ответа также может быть разной: сообщение на экране, звуковой эффект или графическое изображение. В зависимости от ответа программа в соответствии со сценарием опроса выбирает продолжение, осуществляя тем самым «маршрутизацию» опроса (переход к следующему упражнению, переход к более сложной части опроса, возврат к подготовительной части, прекращение опроса и т. д.).

*Программа управления экраном* осуществляет прерывание диалога для выполнения учащимся необходимых вспомогательных действий (например, вычислений, геометрических построений, эксперимента с использованием модели, обращения к справочной системе и т. д.), освобождение экрана с запоминанием текущего номера кадра сценария урока, возврат к продолжению опроса, полнокровное движение по сценарию вперед-назад.

Перейдем к рассмотрению средств программирования.

*Преппроцессор* — программа, предназначенная для расширения системы команд базовой машины путем моделирования их на существующей системе команд. Примеры использования преппроцессора — введение в язык программирования Бейсик средств описания и вызова процедур, описания локальных переменных, структур данных и т. д. Преппроцессор получает на входе текст исходной программы и производит замену предложений, не являющихся командами базовой

машины, последовательностями команд базовой машины. Преобразованная программа передается на трансляцию. Результаты работы преппроцессора скрыты от пользователя, вставляемые им команды не выдаются на листинг.

*Программная поддержка технологии нисходящего проектирования* содержит необходимые средства, позволяющие разработчику программ включать в текст программы сегменты, получаемые в процессе последовательного уточнения; выдавать листинг участка программы с подавлением печати тех сегментов, которые отражают раскрытие отдельных предложений, полученных в ходе нисходящего проектирования; пользоваться имитаторами («заглушками») для реализации процесса нисходящего контроля.

*Отладочные средства* — программы, дополняющие существующие в составе программного обеспечения ЭВМ общесистемные отладочные средства: получение трассировочных таблиц; реализация стартопного выполнения; средства проверки эффективности программ построением профиля исполнения, содержащего листинг, который показывает, сколько раз выполнялась каждая команда и сколько единиц времени затрачено на ее выполнение.

## Функциональное наполнение

В качестве примера рассмотрим ряд программ и моделей, которые можно рассматривать как функциональное наполнение ПОС по предмету «Основы информатики и вычислительной техники». Описание программы состоит из трех основных разделов, которые в совокупности можно назвать паспортом программы, включающим содержание, цель использования, примеры задач, которые можно решать при ее наличии.

## Изучение основ математической логики и теории алгоритмов

1. Операции над функциями алгебры логики: минимизация, преобразование из формального представления в таблицу истинности и обратно.

**Содержание.** Учащийся записывает на экране формулу, являющуюся записью функции алгебры логики. На экране автоматически отображается таблица истинности, соответствующая этой формуле. Обратно — по таблице истинности, задаваемой учащимся, на экране отображается соответствующая функция алгебры логики.

**Цель использования демонстрационной**

модели. Усвоение основных операций алгебры логики — И, ИЛИ, НЕ.

**Примеры задач.** Учащийся, пользуясь данным пакетом, может решать задачи логического характера не умозрительно, а с применением законов математической логики.

2. Имитатор машины Тьюринга. На экране отображается модель машины Тьюринга: лента с информацией, головка записи-считывания, логический блок. Эту модель можно программировать; при выполнении программы имитатор изображает состояние ленты и машины на каждом шаге выполнения программы, например программы сложения двух чисел.

**Цель использования демонстрационной модели.** Усвоение принципов функционирования простейшей ЭВМ и понятия алгоритма, представляемого в виде точного предписания о том, что должна делать машина на каждом такте работы.

**Примеры задач.** Учащимся предлагается самостоятельно составить алгоритм решения задачи на машине Тьюринга. Это значит, что по условию задачи необходимо составить функциональную схему, задать начальное состояние и дать начальную конфигурацию ленты. Указанная информация является входом для имитатора, который учащийся использует для проверки правильности составления алгоритма.

## Выполнение операций над структурами данных

**Содержание.** Учащиеся изучают способы реализации основных операций (запись, чтение, добавление нового элемента, исключение элемента) для следующих структур данных:

- массив;
- связный список;
- стек;
- очередь;
- дерево.

Для каждой из указанных структур на одной половине экрана демонстрируется ее графическое изображение, на другой — способ реализации операций над этими структурами на базовом языке программирования.

**Цель использования демонстрационной модели.** Усвоение способов задания структур данных, используемых для решения разнообразных задач, а также преимуществ и недостатков этих структур с точки зрения затрат времени на выполнение указанных операций; использование массивов (поиск данных по ключам); использование связных списков (представление транспортной сети); использование стека (решение задачи синтаксического разбора скобочных выражений); использование очереди (моделирование



системы обслуживания); использование дерева (представление в машине записи арифметического выражения).

### Реализация поиска и сортировки на упорядоченных структурах данных

**Содержание.** На экране демонстрируется часть последовательного файла (например, телефонного справочника). В каждой записи выделен ключ. По запросу осуществляется сортировка этого файла по выбранному ключу. По другому запросу осуществляется включение новой записи (с раздвижкой файла) или исключение какой-либо записи.

Показывается, каким образом указанный файл представляется в машине в виде массива и сколько времени нужно на поиск и корректировку информации.

Показывается, каким образом указанный файл можно представить в виде дерева, и демонстрируется, что в этом случае на корректировку информации нужно значительно меньше времени, чем в случае массива. На экране изображается конфигурация дерева до и после корректировки.

Демонстрируется внешняя сортировка двух упорядоченных файлов методом слияния. На экране изображаются два исходных файла до слияния и результирующий файл.

**Цель использования демонстрационных моделей.** Усвоение методов поиска и сортировки информации и зависимость затрат времени от способов ее представления.

**Примеры задач.** Разнообразные задачи информационного поиска.

### Реализация численных методов решения задач

**Содержание.** А. Вычисление функций при заданном значении аргумента. Демонстрируются различные кривые в графике высокого разрешения: розетки  $r = a + b \sin(mt)$ ; фигуры Лиссажу  $x = a \sin(mt)$ ,  $y = b \sin[n(t + \alpha)]$ ; «вычурные» кривые: циклоиды, спирали и т. д.

Б. Решение нелинейных уравнений методом последовательных приближений. Уравнение  $f(x) = 0$  представляется в виде  $x = \varphi(x)$ , и на экране в системе координат изображаются графики  $y = x$  и  $y = \varphi(x)$ . После задания начального приближения демонстрируется сходимость (или несходимость) процесса.

В. Численное интегрирование. Учащийся задает уравнение кривой  $y = f(x)$  и границы отрезка  $x_1 = a$ ,  $x_2 = b$ , а также шаг  $h$ . На экране выделяются истинная площадь под кривой и приближенная прямоугольниками. При изменении шага  $h$  показывается, как меняется точность вычисления площади.

Г. Приближение результатов эксперимента

по методу наименьших квадратов. На экране учащийся задает точки в системе координат  $(x, y)$  и режим (линейная или квадратичная аппроксимация). Результатом является кривая наилучшего приближения.

**Цель использования демонстрационных моделей.** Усвоение численных методов решения задач и усвоение понятия параметра (по желанию можно получить на экране параметрическое семейство кривых).

**Примеры задач.** Учащемуся предлагается составить программу для решения задачи, которая использует одну из указанных программ в качестве готовой процедуры. Данный пакет можно использовать в любых задачах, требующих применения численных методов.

### Имитационное моделирование

**Содержание.** Демонстрируется простая система массового обслуживания (например, обслуживание клиентов в мастерской). Есть два процесса: выдача заявки на обслуживание и конец обслуживания. С ними связываются две случайные величины:  $X$  — время прибытия клиента и  $Y$  — длительность обслуживания клиента. Результатом моделирования является получение ответов на следующие вопросы:

число прибывших клиентов;  
средняя длина очереди;  
максимальная длина очереди;  
число клиентов, которым пришлось ждать.

**Цель использования демонстрационной модели.** Усвоение понятия имитационного моделирования и разницы между численным методом решения задачи и методом, основанным на применении имитационных моделей (чаще всего ответом является не численное значение какой-либо величины, а процесс, развивающийся во времени; среди аргументов могут присутствовать случайные величины).

**Примеры задач.** Кроме многочисленных задач из области систем обслуживания можно привести задачи из области охраны окружающей среды (см., например, задачу о кошках, мышах и пшенице в книге С. Гудмана «Введение в разработку и анализ алгоритмов». М.: Мир, 1981).

### Пакет для конструирования игр

**Содержание.** Пакет позволяет учащимся самостоятельно написать программу выбранной ими несложной игры (речь, конечно, не идет о шахматах или шашках).

Программы пакета очень поучительны тем, что демонстрируют методы поиска на структурах данных типа дерева (дерево игры)

и предоставляют возможность учащимся испытать удовольствие поиграть в игру, которую они сами сконструировали.

Пакет имеет 3 стандартные программы, к которым обращается программа, составленная учащимися:

построение очередного яруса дерева игры (вход — массив, содержащий информацию о возможных позициях после очередного хода, выход — структура очередного яруса дерева игры);

прохождение дерева в глубину (вход — текущая вершина дерева игры и структура дерева, выход — следующая вершина дерева);

отсечение ветвей в дереве игры, исходя из оценок вершин (вход — информация о номере яруса, где находится текущая вершина, и оценки позиции в дочерних вершинах; выход — дерево игры, преобразованное путем отсечения ветвей).

Учащийся составляет:

описание игры (описание объектов (фигур), т. е. ключевые названия, цены, правила перемещения при очередном ходе; начальная позиция игры и критерий окончания игры; общие ограничения на возможные позиции; эвристический способ вычисления оценок позиций);

графическое оформление игры (демонстрационная доска, фигуры, протокол выполненных ходов и т. д.).

### Изучение основ вычислительной техники

1. Логическое изучение работы элементов и схем ЭВМ.

**Содержание.** Пользователь с помощью графического редактора конструирует на экране логическую схему, составленную из элемен-

тов И, ИЛИ, НЕ и триггера (элемента памяти). Фиксируются входные полюса схемы и выходные полюса с помощью специальных графических символов. Изучение работы схемы состоит в задании двоичных значений сигналов (0 или 1) на входных полюсах и наблюдении на экране двоичных значений сигналов на выходах элементов. Специальный символ ( $x$ ) идентифицирует неопределенное состояние элементов памяти. Данная модель позволяет определить время переходного процесса в схеме, выявить наличие генерации и т. д. Пользователь может определить влияние изменения связей между элементами на функционирование схемы.

**Цель использования модели.** Усвоение принципов работы элементов и схем ЭВМ с помощью экспериментирования над математической моделью схемы. Особый интерес представляет изучение работы схем сумматоров, умножителей, регистров сдвига, дешифраторов и др.

2. Функциональное моделирование ЭВМ со встроенной программой.

**Содержание.** На экран выводится упрощенная функциональная схема ЭВМ, содержащая арифметическое устройство (процессор), регистры общего назначения, регистр команд, счетчик команд, регистр адреса, индексный регистр и набор ячеек оперативной памяти. Вводится набор команд, выполнение которых реализуется с помощью имитатора: каждая команда моделируется последовательностью операций над регистрами, ячейками памяти и операциями процессора. Выполнение их отображается на экране изменением состояний регистров и ячеек памяти.

**Цель использования модели.** Изучение работы простейшей ЭВМ на логическом уровне и усвоение принципов реализации как отдельных команд, так и программ.

### А. БАРИНОВ

Учебно-методическое управление Госпрофобра СССР

## Компьютер в ПТУ: сегодня и завтра

К началу 1986/87 учебного года в профтехучилищах Российской Федерации введено несколько десятков кабинетов вычислительной техники. К следующему учебному году число их увеличится до нескольких сот, а в 1990 г. каждое второе профтехучилище будет иметь комплект учебной вычислительной техники, состоящий из 12—15 персональных ЭВМ.

Как использовать поступающие средства

вычислительной техники (СВТ) и каким образом внедрять их в учебно-воспитательный процесс? Что изменится при этом в жизни профтехучилища? Как добиться того, чтобы использование компьютеров повысило качество обучения?

Введение предмета «Основы информатики и вычислительной техники», появление в опорных училищах профессиональных ЭВМ типа ДВК-1М и ДВК-2М, первых кабинетов



учебной вычислительной техники становится фундаментом для учебного процесса. Учащиеся делают первый шаг к содружеству с компьютером, ликвидируют психологический барьер в работе с ЭВМ.

Комплексное применение компьютеров в обучении повлечет за собой полную перестройку учебных планов и программ, изменение методики изучения конкретных курсов, порядка и способов подачи и закрепления материала. Разработка экспериментальных компьютерных курсов — это серьезнейшая научная работа.

Комплексное применение компьютера предусматривает его использование на всех стадиях изучения материала:

- в теоретическом обучении;
- в лабораторно-практических работах;
- в производственном обучении.

При теоретическом обучении наиболее целесообразно использовать ПК при закреплении знаний, повторении пройденного материала и контроле его усвоения. Появляется возможность индивидуализировать обучение. Не секрет, что у некоторых учащихся, поступивших в профтехучилища, выявляются серьезные пробелы в знаниях. В этих случаях ПК может стать просто незаменимым. Получив с его помощью основной объем информации, отстающий ученик сможет гораздо быстрее достичь уровня знаний своей группы.

Ценным качеством электронного репетитора является способность наглядно проиллюстрировать на экране изучаемый материал. В этой части ПЭВМ похожи на традиционные ТСО (кино-, диапроектор и т. п.). Однако возможность управлять «картинкой» на дисплее существенно расширяет диапазон применения компьютеров в обучении.

Можно проводить «компьютерные лабораторные работы». Управление «лабораторной установкой», изображаемой на экране дисплея, производится учащимися в диалоге с компьютером.

Применение ЭВМ в обучении показывает, что учащийся не боится проявить незнание и непонимание темы перед компьютером, переспросить его лишний раз. По запросу преподавателя компьютер предоставит протокол лабораторной работы. Это поможет определить ее результативность, подготовленность учащихся. Налицо не только материальная экономия и педагогическая эффективность, но и возможность имитировать уникальные и опасные лабораторные работы. Лабораторная работа по производству серной кислоты разработана учеными Новосибирска для ПЭВМ «Агат» и представляет собой, по сути, модель технологического процесса и в некоторой степени может рассматри-

ваться как программа-тренажер оператора технологической установки.

Важнейшее звено внедрения ПЭВМ в профтехшколе — создание на их базе тренажеров. Стоимость современного станочного оборудования достигает сотен тысяч рублей, и ошибки в управлении технологическими линиями и станками очень дорого обходятся для производства. Поэтому на сегодняшний день целесообразность использования тренажеров в процессе обучения ни у кого не вызывает сомнений.

Тренажеры на базе профессиональных и учебных ПЭВМ можно условно разделить на два класса:

дающие отдельные, начальные умения и знания;

имитирующие реальное оборудование.

Первый класс тренажеров, назовем их программными, представляет собой пакет программ, моделирующих с определенной степенью достоверности работу технологического оборудования.

Специалистами Свердловского инженерно-педагогического института разработана программа такого класса для отработки первичных навыков по работе с токарным станком. Учащийся выбирает форму резца, скорость вращения заготовки, ряд других параметров и на экране наблюдает за ходом обработки. При этом выдаются диагностические сообщения о качестве обработки, нарушениях режимов и др.

Если в программу встроена функция консультации, подсказчика, то ПЭВМ может предложить тот или иной инструмент для обработки или показать, в каком режиме рациональнее ее вести.

Тренажеры данного класса имеют относительно невысокую стоимость, обусловленную лишь стоимостью программного продукта. Простота тиражирования последнего делает такие тренажеры доступными каждому учебному заведению, имеющему соответствующие ПЭВМ. Удельная стоимость тренажера будет тем меньше, чем шире он будет применяться. Да и время его «изготовления», т. е. тиражирования программного продукта, исчисляется минутами.

Второй класс, как правило, требует аппаратной доработки ПЭВМ, создания электронного пульта, имитирующего реальные органы управления станком, установкой. Такой тренажер более полно имитирует реальное оборудование, и его можно назвать тренажером-имитатором. Используя тренажеры-имитаторы до выхода на производственный участок, можно отработать у учащихся сенсомоторные навыки, дать возможность почувствовать реальную работу линии, станка.

В ходе обучения прорабатываются различ-

ные ситуации, вплоть до аварийных, у обучаемого закрепляется более широкий набор реакций, нежели в работе на реальном оборудовании. Это особенно важно при подготовке операторов:

станков с программным управлением;

дистанционного пульта управления в химическом производстве;

установок непрерывной разливки стали и других.

Перечень задач тренажеров обоих классов может быть многократно расширен. Логично возложить на компьютер контроль приобретаемых навыков и управление темпом обучения, тестирование и выявление «белых пятен» в знаниях и навыках учащихся и ряд других задач.

Возвращаясь теперь к реалиям сегодняшнего дня, отметим использование ПЭВМ как инструмента для расчетов, получения информации и другой повседневной работы учащихся. Компьютер — это логарифмическая линейка сегодняшнего дня, учащиеся должны работать с ПЭВМ так же, как со словарем, учебником, справочником.

Компьютером можно и нужно пользоваться при проведении расчетов, получении информации (данных) при подготовке к предстоящей работе в ходе производственного обучения. Базы данных, составленные для производственной деятельности на предприятиях, могут переноситься в СПТУ полностью или несколько адаптированными. Они могут содержать широкий спектр информации:

данные об имеющихся марках стали (пластмасс, сырья, деталей, заготовок и т. д.);

данные о допустимых режимах обработки (резания, сварки, выпечки и т. п.) в зависимости от марки (состава, температуры) исходного материала;

диапазоны допусков в зависимости от качества;

требования ГОСТов и ТУ и ряд других данных.

И еще один аспект широкого внедрения ЭВМ в учебный процесс — это создание АСУ училища. Ведение отчетной документации, планирование учебной работы отнимают значительную часть рабочего времени руководителей, преподавателей и мастеров профтехучилищ.

В реформе школы выдвинуто требование об улучшении условий для творческой работы директоров школ и профтехучилищ по организации учебно-воспитательного процесса.

Приобщение инженерно-педагогических работников училища к работе на ЭВМ заложит основы их компьютерной грамотности, обеспечит более эффективное и научно обо-

снованное руководство учебно-воспитательным процессом среднего профтехучилища.

Создание педагогических программных средств для ПЭВМ, тренажеров, тренажеро-имитаторов, разработка методик их использования, а в дальнейшем создание компьютерных курсов по ряду дисциплин — серьезная и ответственная работа. По оценкам ведущих советских и зарубежных специалистов, для разработки компьютерного урока продолжительностью один час необходимо от 100 до 400 ч работы. В профтехучилищах страны ведется подготовка более чем по 1500 профессиям. Даже выделив из них 100—150 наиболее массовых, перспективных, в подготовке по которым рационально использовать СВТ, можно оценить объем предстоящей работы. Силами системы профтехобразования, ее научных подразделений эту работу не удастся выполнить быстро. Здесь необходимо участие научных организаций АН СССР, вузов страны, отраслевых министерств. С учебными заведениями Минвуза РСФСР такая работа была начата в марте 1985 г. Совместным приказом Минвуза РСФСР и Госпрофобра РСФСР были созданы 22 зональных центра по внедрению вычислительной техники в учебный процесс вузов, техникумов и средних профессионально-технических училищ РСФСР.

В декабре 1985 г. на совместной коллегии был принят комплексный план мероприятий Минвуза РСФСР и Госпрофобра РСФСР на двенадцатую пятилетку, определяющий не только общие направления деятельности зональных центров, но и вполне конкретные разработки. Первые результаты совместной деятельности уже налицо. В Ленинграде, Москве, Свердловске создаются фрагменты компьютерных уроков по конкретным курсам спецдисциплин, общетехническим предметам. Сотрудники вузов участвуют в работе секций преподавателей основ информатики и вычислительной техники, в разработке плакатов, учебно-наглядных пособий, методических рекомендаций. Силами Томского политехнического института и Томского государственного университета издано несколько методических рекомендаций (по ведению отдельных тем основ информатики и вычислительной техники, по технологии вычислений и программирования на микрокалькуляторах), которые успешно используются преподавателями профтехучилищ. Такая работа проводится и в других зональных центрах республики.

Интеграция деятельности научных учреждений высшей и профессионально-технической школы может и должна дать ощутимые результаты.



## Опыт ускоренного обучения

В данной статье описан опыт ускоренного обучения учащихся девятых классов общеобразовательной школы № 117 Брежневского района Москвы по курсу «Основы информатики и ВТ» с использованием ЭВМ по специально разработанной программе.

Следует отметить, что специального отбора в эти классы не проводилось.

### Программа курса и ее обоснование

28

Утвержденные к настоящему времени Министерством просвещения СССР программа курса «Основы информатики и ВТ» и учебник для девятых классов рассчитаны на обучение старшеклассников в период перехода к массовой компьютеризации школ. Этот курс направлен на ознакомление учащихся с основными структурами алгоритмов и этапами решения задач на ЭВМ. Он предполагает отсутствие в школе ЭВМ.

Но для школ, имеющих ВТ, необходима другая учебная программа. Она, так же как и ее безмашинный вариант, должна давать общее представление о предмете и в то же время развивать элементарные навыки работы на современных ЭВМ с использованием существующих языков программирования. Для решения такой задачи был предложен курс, программа которого приведена в прил. 1.

Данная программа является универсальной с точки зрения типа ЭВМ и конкретного языка программирования. Основное место в ней уделяется освоению навыков практической работы, на которую отводится  $\frac{2}{3}$  учебного времени.

Остановимся более подробно на содержании разделов программы и дадим некоторые методические рекомендации.

1. Первый раздел программы посвящен кратко изложению назначения основных блоков современных ЭВМ, знакомству с рабочим местом, с возможностью редактирования текстов и развитию навыков работы с клавиатурой.

Основное место в этом разделе отводится демонстрации прохождения заданий через ЭВМ. Учащиеся должны понять, как будут обрабатываться их будущие программы и зачем нужны те или иные действия. Это необходимо для того, чтобы школьники не ждали чего-либо сверхъестественного от компьюте-

ра и в то же время могли полнее использовать его возможности.

Конкретное знакомство учащихся с ЭВМ начинается с изучения устройств ввода-вывода, имеющихся в учебном классе. Здесь школьники получают навыки работы с клавиатурой дисплея, в том числе с назначением некоторых управляющих клавиш.

Очень серьезно следует подойти к изучению учащимися возможностей по созданию и редактированию текстов. На это рекомендуется отводить половину учебного времени первого раздела, что позволит школьникам в дальнейшем не терять время при исправлении допущенных в программах ошибок. Представляется целесообразным и в дальнейшем на каждом последующем занятии в терминальном классе обращать внимание учащихся на эффективность использования редактора текстов.

2. Перед переходом к усвоению основных структур языка программирования полезно выделить несколько уроков на правила составления алгоритмов для ЭВМ. Обычно учащиеся затрудняются написать правильную программу из-за отсутствия четкого представления об алгоритме решения задачи. Любая формализованная запись процесса решения задачи — будь то блок-схема, словесное описание или алгоритмическая нотация, используемая в учебнике, — должна помочь в преодолении трудностей.

В разделе следует разобрать алгоритмы задач, близких к тем, которые будут затем решаться на ЭВМ. В течение всего курса не стоит забывать о необходимости предварительного разбора и анализа алгоритмов задач, предлагаемых учащимся.

3. Изучение языка программирования начинается знакомством с операциями ввода-вывода. При этом решаются простейшие задачи, демонстрирующие их возможности. Важно с самого начала заинтересовать школьников, подбирая им индивидуальные задания с максимальным использованием графических возможностей терминалов. Одновременно с этим продолжается работа учащихся по закреплению навыка работы с редактором текстов на конкретных примерах — их программах.

В этом же разделе целесообразно познакомить учащихся с имеющимися стандар-

тными функциями языка программирования, которые могут облегчить решение многих учебных задач, как текстовых и графических, так и вычислительных.

Другим важным моментом, требующим внимания на протяжении всего курса, является выработка культуры оформления программы — оформление заголовков, комментариев и т. д.

4. При изучении материала четвертого раздела особое внимание следует обратить на логический анализ задачи и воплощение его результатов в конкретных управляющих операторах — условных, перехода и др. Серьезные сложности возникают у учащихся, когда необходимо правильно организовать переход на ту или иную (в зависимости от проверяемых условий) часть программы.

Для этой темы полезно использовать задачи, содержащие повторяющиеся действия. Это поможет подвести учащихся к понятию цикла и позволит в разд. 5 вернуться к этим задачам, решая их с использованием оператора цикла.

Весьма существенным в обучении является усвоение учащимися приемов отладки программ и проверки их правильности. Необходимо научить школьников следить за последовательностью этапов выполнения программ, например путем применения контрольной выдачи промежуточных результатов на устройства вывода.

Школьники должны научиться подбирать конкретные примеры для проверки правильности работы программ и уметь анализировать полученные результаты. Целесообразно также научить учащихся самим последовательно выполнять операторы программ (вместо компьютера) с конкретными данными — числами, текстовыми величинами.

Освоение навыков отладки дает возможность учащимся самим контролировать правильность своих программ, уменьшит их растерянность при диалоге с ЭВМ. Эффективность занятий в терминальном классе возрастет, а роль преподавателя при этом сведется к консультациям по особо трудным вопросам.

5. Наибольшие сложности при изучении этого раздела возникли у учащихся школы № 117 в понимании параметра цикла не только как счетчика числа повторений, но и как индекса элемента массива, переменного в выражении и т. д.

Так же, как и при изучении других разделов, не следует забывать о наглядности выбираемых задач. Например, введение понятия массива полезно демонстрировать с привлечением образных примеров (месяцы года как элементы массива и пр.).

6. Шестая тема носит познавательный ха-

рактер. Здесь следует дать понятие подпрограммы и правила работы с ней без широкого использования при решении учебных задач. Важность этой темы в полной мере проявится при профессиональном обучении школьников и организации их коллективной работы.

### Подбор задач

Эффективность учебного процесса во многом определяется характером решаемых задач, их доступностью и наглядностью.

1. При освоении редактора текстов (тема № 1) представляется оправданным занятие в терминальном классе проводить в виде машинного диктанта, в процессе которого учащиеся на слух воспринимают текст, зачитанный учителем, заносят его в память ЭВМ и затем исправляют допущенные ошибки.

2. Задачи по второй теме должны охватывать четыре основных типа алгоритмов: линейные, ветвящиеся, циклические и вспомогательные. Разбор задач должен быть направлен на ознакомление с основными этапами решения, а также особенностями алгоритмов и способами их записи.

3. Наиболее характерными задачами для данного типа алгоритмов — линейного — являются простейшие вычислительные задачи, широко встречающиеся в школьных дисциплинах.

4. Достаточно наглядными при изучении ветвящихся алгоритмов являются задачи поиска максимума из трех и более чисел, а также программы, планирующие свою работу в зависимости от введенных в режиме диалога значений параметров.

5. Повышенный интерес у школьников неизменно вызывает составление программ, рисующих картинку с учетом графических возможностей устройств вывода. Такие задачи могут быть решены с использованием линейных структур языка программирования, но наибольший эффект достигается при реализации их на основе циклических алгоритмов. В дальнейшем такие программы будут служить основой для освоения аппарата вспомогательных программ — подпрограмм.

После освоения оператора цикла можно дополнительно разобрать задачи, связанные с обработкой текста и массивов.

Ряд интересных задач предложен в книге А. И. Салтыкова и Г. Л. Семашко «Программирование для всех» (М.: Наука, 1985). Там же просто и доступно изложены некоторые рекомендации по освоению языка программирования.

### Организация учебного процесса и его результаты

Ученики девятых классов школы № 117

29



были разделены на четыре группы по 15—16 человек каждая. Обучение производилось в первом полугодии раз в неделю в течение 6 ч: 1 ч — за счет курса информатики, 4 ч — трудовое обучение, 1 ч — факультатив. Треть из них пришлось на занятия в терминальном классе. Общий объем курса составил 102 ч. Занятия проводились с использованием языка программирования Бейсик на мини-ЭВМ и персональных компьютерах в терминальных классах на 16 учебных мест.

В конце курса был проведен экзамен, результаты которого выявили условно три группы школьников. В первую группу — она составила примерно 25 % — вошли учащиеся, умеющие быстро и самостоятельно построить алгоритм решения предложенной задачи и реализовать его на языке программирования. Подавляющее большинство школьников этой группы учатся на «хорошо» и «отлично» по остальным школьным предметам.

30 Вторую группу — самую многочисленную, примерно 50 %, — составили учащиеся, освоившие структуры языка программирования, но испытывающие затруднения при самостоятельной разработке алгоритмов.

Оставшиеся — 25 % — проявили равнодушие к предложенному курсу, несмотря на тщетные попытки преподавателей заинтересовать их индивидуальными, специально подобранными заданиями. Интересно, что половина школьников из этой группы в начальный период обучения вынуждена была заниматься на ЭВМ, у которых 50 % учебного времени тратилось на приведение их в рабочее состояние (из-за низкой надежности).

Приложение 1. Программа курса ускоренного обучения основам информатики и вычислительной техники в IX классах школы № 117 Брежневского района Москвы.

Тема 1. Устройство и принципы работы ЭВМ — 12 ч (из них в терминальном классе — 5).

1.1. Возможности ЭВМ — 1.

1.2. Общая схема устройства ЭВМ, принципы их работы, система математического обеспечения — 2.

1.3. Знакомство с устройствами ввода-вывода; получение навыков работы с клавиатурой — 1 (1).

1.4. Редактирование текстов на ЭВМ — 6 (4).

1.5. Представление информации в ЭВМ. Двоичная система счисления — 2.

Тема 2. Понятие алгоритма — 4.

2.1. Понятие алгоритма; данные и операции над ними — 1.

2.2. Способы записи алгоритмов, блок-схема; структура программы — 1.

2.3. Составление простейших алгоритмов — 2.

Тема 3. Введение в язык программирования; линейные алгоритмы — 25 (10).

3.1. Основные понятия и символы языка программирования — 1.

3.2. Типы данных и операторы спецификаций. Форма записи чисел; переменные, массивы, арифметические операции — 2.

3.3. Стандартные функции: математические, строковые и графические — 6 (3).

3.4. Операторы ввода-вывода — 10 (4).

3.5. Переменные, арифметические и логические выражения, оператор присваивания — 6 (3).

Тема 4. Ветвящиеся алгоритмы — 18 (6).

4.1. Управляющие операторы — 6 (3).

4.2. Принципы отладки и тестирования программ. Составление простейших программ на языке программирования (решение вычислительных задач, диалоговые программы, обработка текста) — 12 (3).

Тема 5. Циклы и массивы — 15 (5).

5.1. Циклические процессы, оператор цикла — 6 (2).

5.2. Решение задач с использованием массивов — 9 (3).

Тема 6. Основы структурного программирования — 18 (4).

6.1. Подпрограммы — 4 (2).

6.2. Модульная структура программ. Библиотеки подпрограмм и их использование — 4 (2).

6.3. Практика решения задач на ЭВМ — 10 (4).

# Методика обучения

## Методические указания по преподаванию курса «Основы информатики и вычислительной техники» в X классе\*

Язык программирования Рапира (6 ч)

§ 10. Запись алгоритмов в виде процедур на Рапире (1 ч)

Основная цель. Познакомить учащихся с представлением алгоритмов в языке программирования Рапира, символикой языка и синтаксисом основных конструкций языка.

Требования к знаниям и умениям. Учащиеся должны уметь описывать простые алгоритмы, знать обозначения основных конструкций и операций и записывать с их помощью арифметические выражения.

Методические указания. Содержанием урока является обсуждение примера простой программы на Рапире путем сравнения ее с описанием соответствующего алгоритма на алгоритмическом языке. Рекомендуется описание алгоритма на алгоритмическом языке предварительно записать на доске, а описание процедуры на Рапире вводить последовательно, обсуждая строки примера.

Язык программирования Рапира позволяет описывать сложные действия, в том числе все те, которые были показаны учащимся при знакомстве с алгоритмическим языком. Кроме того, многие конструкции Рапиры и алгоритмического языка либо совпадают, либо весьма близки. Поэтому аналогию этих двух средств описания алгоритмов можно использовать в качестве основного инструмента в методике обучения программированию на языке Рапира.

Изучение каждого языка, в том числе и языка программирования, начинается с его алфавита. Язык программирования, предназначенный для общения человека с машиной, выполняющей написанную им программу, составляется не из произвольных символов, а только из тех, которые имеются на клавиатуре ЭВМ. В силу того что на клавиатуре можно набирать только строчные, «линейные», выражения, в языках программирования существуют специальные способы обозначения подстрочных индексов и надстрочных показателей, широко применяемых в математике и используемых в алгоритмическом языке.

Нет на клавиатуре ЭВМ и таких знаков алгоритмического языка как  $\sqrt{\quad}$  — квадратный корень или  $||$  — вертикальные черты, используемые для обозначения абсолютного значения, нет греческих букв. Все эти особенности учитываются в языках программирования.

На этом уроке школьники знакомятся с алфавитом Рапиры, в который входят все русские и латинские буквы, десять цифр (от 0 до 9) и набор специальных символов.

Многие из этих символов могут встречаться в программах в нескольких различных значениях.

Знаки арифметических операций отличаются от принятых в математике: умножение обозначается звездочкой \*, которую не разрешается опускать там, где в математике это допускается: в программировании следует писать, например, 2\*A (2A писать запрещено!); вообще ни в одной из операций, записываемых в арифметических выражениях на языке программирования, нельзя опускать знаки операций;

\* Окончание. Начало см.: Информатика и образование №№ 1, 2



деление обозначается наклонной чертой / (никакие другие обозначения деления недопустимы, частное следует записывать A/B);

возведение в степень имеет обозначение \*\* (например, куб суммы квадратов  $(a^2+b^2)^3$  надо записать так:  $(A**2+B**2)**3$ ).

Поупражняйтесь в записи арифметических выражений на языке Рапира: запись квадратного трехчлена  $ax^2+bx+c$  на языке программирования имеет вид  $A*X**2+B*X+C$ ;

выражение силы в формуле закона Кулона  $k \frac{Q_1 \cdot Q_2}{R^2}$  программист должен записать так:  $K*Q1*Q2/R**2$ .

Заметьте, что «линейность» записи арифметического выражения может потребовать лишних скобок; выражение  $\frac{a}{b \cdot c}$  (без скобок) нельзя в программе записать  $A/B \cdot C$ , потому что это означает произведение частного  $A/B$  на множитель  $C$  (в силу обычного порядка операций одинакового приоритета — слева направо), т. е.  $\frac{a}{b} \cdot c$ ; чтобы получить желаемый результат, надо писать  $A/(B \cdot C)$ .

Объясняя ученикам правила записи арифметических выражений на примере приводимого в учебнике описания процедуры, учитель должен после этого рассмотреть и другие, более выразительные в этом отношении, примеры. Так, можно математическую запись выражения

$\frac{217-34}{(2^3-6) \cdot 2^5 \cdot (3^2-2)}$  — представить в Рапире следующим образом:

$(217-34)/(2**3-6)/(2**5*(3**2-2))$ .

Необходимо снять все вопросы, которые могут возникнуть у школьников при обсуждении записи такого выражения, и зафиксировать эту запись в тетради; она поможет учащимся в выполнении домашнего задания, включающего еще один подобный пример.

В арифметических выражениях используются математические функции, для которых в языке программирования нет специальных символов. Для их обозначений используются служебные слова. Примером такой функции в обсуждаемой на уроке программе может служить квадратный корень. Его обозначение  $SQRT(X)$ . Например, расстояние между двумя точками  $(x1, y1)$  и  $(x2, y2)$  можно вычислять как значение выражения  $SQRT((X2-X1)**2+(Y2-Y1)**2)$  (одно из домашних упражнений состоит в записи такого выражения).

Вот принятые в Рапире обозначения некоторых других часто используемых (стандартных) функций:

SIN(X) — синус,  
COS(X) — косинус,  
TG(X) — тангенс,  
ARCSIN(X) — арксинус,  
ARCCOS(X) — арккосинус,  
ABS(X) — абсолютное значение,  
EXP(X) — экспонента,  
LG(X) — десятичный логарифм,  
LN(X) — натуральный логарифм.

Служебные слова Рапиры, имеющие, как и служебные слова алгоритмического языка, фиксированное начертание и смысл, по существу являются элементами алфавита. Часть этих слов повторяет лексику алгоритмического языка (ЕСЛИ, ТО, ИНАЧЕ, ВСЕ, ПОКА, НАЧ, КОН). С некоторыми другими служебными словами Рапиры учащиеся знакомятся на уроках информатики (в том числе и на сегодняшнем): ПРОЦ, ИМЕНА. Служебные слова в Рапире не принято подчеркивать.

Поскольку на клавиатуре многих ЭВМ отсутствуют малые строчные буквы, принято записывать заглавными буквами как служебные слова, так и имена (полагая, что алфавит Рапиры включает только заглавные русские и латинские буквы).

Правила записи текстов в Рапире полностью совпадают с соответствующими правилами алгоритмического языка: тексты в программах окаймляются кавычками. А для имен (как для имен переменных величин, так и для имен процедур, являющихся названиями алгоритмов) принята условность; объясняемая и иллюстрируемая в учебнике: слова, составляющие одно многословное имя, соединяются знаком подчеркивания. Например, имя, которое в алгоритмическом языке записывается так:

в Рапире должно быть записано: ПУТЬ\_ДО\_ЛУНЫ.

Описание процедуры начинается с ее заголовка. Основное отличие заголовка описания процедуры (в языке программирования) от заголовка описания алгоритма (в алгоритмическом языке) состоит в том, что в Рапире не требуется описывать типы используемых величин. Распознавание типов тех или иных величин машина осуществляет непосредственно перед выполнением программы. Ученик, получая такую «свободу», должен тем не менее строго следить, чтобы в одной операции не участвовали операции разных типов.

В Рапире иначе, чем в алгоритмическом языке, обозначается роль величин как аргументов и результатов. Она отмечается не отдельными строками описания, а непосредственно в скобках заголовка, и не служебными словами алг и рез, а комбинациями специальных символов: стрелка  $=>$  перед именем переменной означает аргумент (или входной параметр); стрелка  $=>$  после имени переменной означает результат (или выходной параметр).

Совокупность перечисляемых в заголовке описания процедуры имен (как аргументов, так и результатов) называют параметрами описания процедуры, или еще формальными параметрами. Выражения же, подставляемые на место параметров описания при каждом вызове процедуры, называют параметрами вызова, или фактическими параметрами.

Процедуры могут иметь в числе своих формальных параметров переменные, которые играют одновременно роль и аргументов, и результатов. Такие переменные отмечаются в списке формальных параметров двумя стрелками: одна — перед именем переменной, а другая — после него, например:  $=>X=>$ .

Следует заметить, что стрелку входных параметров (предшествующую имени переменной) разрешается опускать при описании процедур. Машина, обнаружив в описании процедуры параметр без стрелки, автоматически, по умолчанию, воспринимает такую переменную как входной параметр — аргумент. Это соглашение, используемое программистами, позволяет писать более компактные заголовки описаний процедур. Хотя немногочисленные примеры учебника не используют этого принимаемого по умолчанию соглашения, учитель может предложить это сокращение ученикам, но, конечно, не на первом уроке по Рапире, а только после того, как учениками достаточно прочно усвоена роль формальных параметров и их обозначения.

То же замечание надо сделать и относительно служебного слова ИМЕНА, которое концентрирует в начале описания процедуры перечисление всех используемых в процедуре величин. В алгоритмическом языке тоже надо было описывать все промежуточные величины, однако место этих описаний не фиксировалось, и, в частности, при описании алгоритмов учащиеся вводили описания промежуточных величин непосредственно перед их первым упоминанием в описании алгоритма. В Рапире разрешено опускать слово ИМЕНА со списком промежуточных величин. Оно становится необходимым тогда, когда описываются достаточно сложные программы, в которых одна процедура содержит многоэтажные вызовы других процедур — вспомогательных алгоритмов. В таких случаях список, записанный после служебного слова ИМЕНА, позволяет машине четко различать имена переменных, используемых только внутри одной процедуры (локальные имена), и переменных, используемых под одним и тем же именем во всех процедурах программы (глобальные имена). Таким образом, запись процедуры приведенного примера имеет вид, показанный в § 10 учебника на с. 58.

Учитель на этом уроке может не объяснять школьникам, как работают используемые в этом примере команды присваивания и условная команда. Эти команды знакомы учащимся по алгоритмическому языку, где они записываются в таком же начертании. Точно так же совпадают с синтаксисом алгоритмического языка такие команды Рапиры, как цикл со служебным словом ПОКА, цикл со служебным словом ДЛЯ или многозначное ветвление (команда, начинающаяся служебным словом ВЫБОР). Например, учитель может предложить интересующимся ученикам описать процедуру, вычисляющую значение суммы  $1+1/2+1/3+...+1/100$ .

Здесь очевидна необходимость цикла. Если обозначить переменную цикла (знаменатель суммируемых дробей) через N, то легко видеть, что ее начальное значение равно 1, конечное значение равно 100, шаг — единице. Следовательно, описание процедуры может иметь вид:

ПРОЦ РЯД (СУММА  $=>$ ).

ИМЕНА: N

НАЧ

N:=1; СУММА:=1

ПОКА N<=100

НЦ



SUMMA:=SUMMA+1/N  
N:=N+1

КЦ

КОН

Если вместо цикла со словом ПОКА использовать цикл со служебным словом ДЛЯ, то ту же процедуру можно переписать:

ПРОЦ РЯД (СУММА =>)

ИМЕНА: N

НАЧ

SUMMA:=1  
ДЛЯ N ОТ 1 ДО 100

НЦ

SUMMA:=SUMMA+1/N

КЦ

КОН

Завершив обсуждение примера процедуры РЕШЕНИЕ\_КВАДРАТНОГО\_УРАВНЕНИЯ, предложите учащимся выполнить в классе упражнения этого параграфа № 1 и 2. Первое из них выносится на доску, а результат фиксируется в классных тетрадях учащихся. Второе — устное; его учитель может сформулировать в виде вопросов к двум учащимся (один отвечает об аргументах, другой — о результатах процедуры ПРИМЕР).

Домашнее задание: § 10, № 3—6.

Указания к решению задач.

1.  $B**3 + \text{SQRT}(C) - \text{SQRT}(A)$

2. В процедуре ПРИМЕР аргументами служат переменные A и C, а результатами — B и S.

3.  $\text{ВЕЛИЧИНА} := (1024/2**5 + 128/4**2) / ((16 - 2*4) - 2**2)$

4.  $N := N - O + 9.81 * T**2 / 2$

Здесь в константе ускорения свободного падения в качестве десятичного разделителя поставлена не запятая, а точка. Это не ошибка, а правило языка программирования. Однако сейчас можно не акцентировать на этом внимание учеников. Пояснение предлагается сделать им на одном из последующих уроков.

5. Решение рассмотрено выше, в комментариях к уроку.

6. ПРОЦ ПРОВЕРКА ( $=>P, =>Q=>R, Z=>$ )

ИМЕНА: X, Y, Z

НАЧ

РЕШЕНИЕ\_КВАДРАТНОГО\_УРАВНЕНИЯ (P, Q, R, X1, X2, Y)

ЕСЛИ Y= «НЕТ РЕШЕНИЙ»

ТО Z:= «ВЕЩЕСТВЕННЫХ КОРНЕЙ НЕТ»

ИНАЧЕ

ЕСЛИ  $X1 * X2 < 0$

ТО Z:= «КОРНИ ПРОТИВОПОЛОЖНЫХ ЗНАКОВ»

ИНАЧЕ Z:= «КОРНИ ОДНОГО ЗНАКА»

ВСЕ

ВСЕ

КОН

Это задание сложнее предыдущих: в нем фигурирует команда вызова процедуры РЕШЕНИЕ\_КВАДРАТНОГО\_УРАВНЕНИЯ. До сих пор на уроке команда вызова процедуры не демонстрировалась. Однако неоднократно подчеркиваемая аналогия с алгоритмическим языком должна подсказать учащимся, что ею можно воспользоваться так же, как мы уже много раз выполняли вызовы вспомогательных алгоритмов. Эту рекомендацию учитель должен дать учащимся, объясняя содержание домашнего задания. И тем не менее следующий урок полезно начать с обсуждения этого последнего домашнего задания.

§ 11. Запись алгоритмов вычисления значения функций на Рапире (2 ч)

Учителю рекомендуется следующее распределение материала параграфа: 1-й урок — правила описания и вызова процедур, вычисляющих значения функций в Рапире; 2-й — решение упражнений на использование функций в программах, самостоятельная работа.

Основная цель. Ввести средства описания алгоритмов вычисления функций в Рапире и закрепить навыки описания и использования функций в программах.

Требования к знаниям и умениям. Учащиеся должны отчетливо выделять те из процедур, действие которых сводится к вычислению единственного результативного значения; они должны уметь программировать такие действия, оценивая преимущества процедур-функций.

Методические указания. При изучении материала параграфа следует опираться на уже усвоенный ранее материал — алгоритмы вычисления функций в алгоритмическом языке.

11.1. В начале урока учитель повторяет с классом материал прошлого занятия: проводит обсуждение одного из упражнений домашнего задания, в ходе которого следует еще раз выяснить, в какой мере усвоен важнейший навык, формируемый в данном курсе, — вызов вспомогательных алгоритмов и, в частности, вызов процедур, когда речь идет о составлении программ на Рапире. Поскольку необходимо проверить домашнее задание, целесообразно попросить одного из учеников написать решение на доске и объяснить его. В таком объяснении важно выделить следующие моменты:

вызов вспомогательного алгоритма (вызов процедуры) представляет собой отдельную команду;

при вызове процедуры на месте формальных параметров в заголовке процедуры записываются конкретные значения — параметры вызова;

в общем случае параметры вызова (фактические параметры), являющиеся аргументами (входные), могут иметь вид любого арифметического выражения; в примере из домашнего задания все три входных параметра представляют собой имена переменных — P, Q, R, но можно себе представить вызов процедуры РЕШЕНИЕ\_КВАДРАТНОГО\_УРАВНЕНИЯ и в таком виде:

РЕШЕНИЕ\_КВАДРАТНОГО\_УРАВНЕНИЯ (5—2\*\*2, 1+4, 2 — —3, M, N, L);

на месте результатов (выходных параметров) в вызове процедуры могут стоять только имена результатов;

имена формальных параметров и соответствующих им параметров вызова (фактических параметров) не обязательно совпадают.

Завершив повторение, учитель переходит к изложению основного материала урока. Процедуры, вычисляющие значения функций, представляют собой частный, но весьма важный как в практическом, так и в методическом отношении вид процедур. По своему содержанию это понятие полностью эквивалентно изученному ранее в алгоритмическом языке понятию алгоритма вычисления функции. Поэтому и этот урок, так же как и предыдущий, построен на сравнении двух программ, одна из которых написана на алгоритмическом языке, а другая — на Рапире.

Сравнение этих двух описаний позволяет дать представление о способах описания функций в Рапире. Как было показано на предыдущем уроке, в Рапире нельзя воспользоваться указаниями типов при описании величин, в том числе и значений функций. Для описания функций это оказывается существенным: в алгоритмическом языке тип вырабатываемого функцией результата служит признаком, по которому алгоритм, вычисляющий значение функции, отличается от всех других алгоритмов. Как поступить в Рапире в этом случае?

С целью выделения функций как самостоятельных объектов языка их описания начинают не служебным словом ПРОЦ (как для обычных процедур), а специально для этого используемым служебным словом ФУНК (сокращение от слова «функция»).

В Рапире, как и в алгоритмическом языке, для выделения единственного выработанного процедурой-функцией результата пользуются не записываемыми в заголовке признаками *arg* (в алгоритмическом языке) или исходная стрелка  $=>$  (в обычных процедурах Рапиры), а специальным присваиванием, в левой части которого стоит имя стандартной переменной ЗНАЧ. Выражение, указанное в правой части этого присваивания, и определяет результат функции. Таким образом, в заголовке описания процедуры, вычисляющей функцию, всегда присутствуют только входные параметры — аргументы. А на прошлом уроке было показано, что стрелки входных параметров в заголовке описания можно опускать, так как параметры без стрелок по умолчанию воспринимаются машиной как входные. Поэтому у процедур, вычисляющих значения функций, заголовки описаний, как правило, не имеют стрелок.

В этом параграфе на примере функции ПЛОЩАДЬ показан полезный прием хорошего стиля программирования — комментирование программ. Включение комментариев в программу делает ее удобочитаемой. Прокомментированную контрольную работу или домашнее задание учитель сможет проверить легче и быстрее. Поэтому он рекомендует этот прием ученикам еще и в интересах повышения эффективности своего труда.

Далее учитель должен предложить учащимся задачи, в которых используются вызовы рассмотренных на уроке функций.

Пример 1. Присвоить переменной величине ДЕЛИТЕЛЬ максимальный из двух наиболь-



ших делителей: один для пары числовых значений АЛЬФА и БЕТА, а другой — для пары ГАММА и ДЕЛЬТА.

Решение дается командой

ЕСЛИ НОД (АЛЬФА, БЕТА) > НОД(ГАММА, ДЕЛЬТА)

ТО ДЕЛИТЕЛЬ:= НОД (АЛЬФА, БЕТА)

ИНАЧЕ ДЕЛИТЕЛЬ:= НОД(ГАММА, ДЕЛЬТА)

ВСЕ

В частности, если переменным присвоены значения  
АЛЬФА:=28; БЕТА:=8; ГАММА:=24; ДЕЛЬТА:=18,  
то величина ДЕЛИТЕЛЬ получит значение 6, равное НОД (24, 18).

Пример 2. Вычислить площадь параллелограмма, составленного из двух треугольников. Длины сторон параллелограмма обозначены Р и Q, а диагональ — R. С помощью описанной на уроке функции ПЛОЩАДЬ (А, В, С) площадь S параллелограмма вычисляется по команде  
S:= ПЛОЩАДЬ (Р, Q, R)

По этой команде значения Р, Q, R подставляются в качестве параметров в процедуру-функцию ПЛОЩАДЬ. Для них процедура вычисляет площадь треугольника со сторонами Р, Q, R, затем полученный результат удваивается.

На этих примерах учитель обращает внимание школьников, что процедуры, вычисляющие значения функций, используются в программах так же, как обычные математические функции в арифметических выражениях.

36 11.2. На втором уроке по теме решаются задачи. Цель этого урока состоит в закреплении навыков работы с функциями в Рапире — их описаниями и вызовами.

Полезное упражнение — показать, как программа, знакомая учащимся в виде обычной процедуры, может быть описана в виде функции. Рассмотренная ранее процедура РЯД имеет описание

ПРОЦ РЯД(СУММА =>)

ИМЕНА: N

НАЧ

N:=1

ПОКА N<=100

НЦ

СУММА:=СУММА+1/N

N:=N+1

КЦ

КОН

Ее можно представить процедурой-функцией без параметров:

ФУНК РЯД

В заголовке отсутствует имя результата, нет его и в самом описании: роль его играет стандартная переменная ЗНАЧ.

ФУНК РЯД

ИМЕНА: N

НАЧ

N:=1

ПОКА N<=100

НЦ

ЗНАЧ:=ЗНАЧ+1/N

N:=N+1

КЦ

КОН

Если количество суммируемых дробей не фиксировать, а попытаться сделать на основе этой процедуры-функции некоторую другую, которая сможет вычислить сумму последовательности любого (заданного) числа L дробей, то у такой функции появится входной параметр (аргумент) L:

ФУНК РЯД (L)

ИМЕНА: N

НАЧ

N:=1

ПОКА N<=L

НЦ

ЗНАЧ:=ЗНАЧ+1/N

N:=N+1

КЦ

КОН

С помощью функции РЯД (L) легко вычислить сумму дробей, являющихся любым отрезком ряда  $1+1/2+1/3+...+1/N+...$ , например, от 15-го его члена до 28-го. Для этого достаточно записать на Рапире одну команду присваивания, в правой части которой дважды вызывается функция РЯД:

ОТРЕЗОК:= РЯД(28)—РЯД(15)

Еще один пример процедуры-функции поможет учащимся легче воспринять одно из упражнений домашнего задания и, кроме того, напомнить учащимся команду многозначного ветвления ВЫБОР. При знакомстве с Рапирой эта команда еще не упоминалась.

Пример. Написать процедуру-функцию ОТМЕТКА, которая по задаваемому значению целочисленного параметра от 2 до 5 вырабатывает текстовое значение названия соответствующей оценки. Выяснив, насколько учащиеся помнят команду многозначного ветвления, можно попросить одного из учеников сделать это описание непосредственно на доске с обсуждением записанного всем классом.

ФУНК ОТМЕТКА (N)

НАЧ

ВЫБОР

ПРИ N=2:ЗНАЧ:=«ПЛОХО»

ПРИ N=3:ЗНАЧ:=«УДОВЛЕТВОРИТЕЛЬНО»

ПРИ N=4:ЗНАЧ:=«ХОРОШО»

ПРИ N=5:ЗНАЧ:=«ОТЛИЧНО»

ВСЕ

КОН

Значение процедуры-функции ОТМЕТКА (N) — текстовое, следовательно, к нему можно применять любые операции над текстами. Так, имеет смысл команда

БУКВА:=ОТМЕТКА (15—12) [1]

Здесь сначала вычисляется значение фактического параметра вызываемой процедуры-функции  $15-12=3$ , а затем вычисляется значение функции при параметре вызова, равном 3. Наконец, у вычисленного результата-текста вырезается первый символ, который и присваивается переменной БУКВА. В нашем примере ответом является односимвольный текст У.

После упражнений, показавших школьникам правила вызова функций в программах, можно дать определение стандартной (или встроенной) функции языка программирования.

Эти функции программисты используют в фиксированном смысле. Они имеют стандартные имена, зная которые, программист может вызывать соответствующие функции, не создавая предварительно их описания. Стандартные функции являются частью языка программирования, встроены в него. Нам уже встречались стандартные функции Рапиры: SIN (X), SQRT (X), ABS (X) и т. д. Такой же стандартной функцией является ДЛИН (X); ее параметром служит текст, а результирующим значением — длина текста (количество символов). Определение стандартной функции не приводит учащихся к новым объектам языка, но «проливает свет» на происхождение непривычного в математике обозначения обращения к функции.

В заключение урока учитель задает классу вопрос для устного ответа. Он пишет на доске строку

A (X); C:=B(X)

и, сказав, что речь идет о некотором фрагменте незнакомой программы, спрашивает, какая из названных в этом фрагменте процедур — A или B — вычисляет значение функции. Ответить на этот вопрос легко, если вспомнить, что вызов обычной процедуры — это команда, а вызов функции — значение.

В конце урока учитель проводит со школьниками небольшую самостоятельную работу, в которой школьники должны оформить в виде процедур, вычисляющих значения функций, известные им из программы IX класса описания алгоритмов.

Для того чтобы сосредоточить внимание и усилия учащихся на главном вопросе работы — правилах описания процедур, вычисляющих функцию, и их вызова, — учитель напоминает им алгоритмы A 10 и A 11 из библиотеки алгоритмов (см. ч. 1 учебника, с. 94, 95).



# Самостоятельная работа (15 мин)

## Вариант I.

Воспользовавшись алгоритмом А 10 из библиотеки алгоритмов, постройте описание процедуры-функции, вычисляющей значение площади криволинейной трапеции, ограниченной прямыми  $X=A$ ,  $X=B$ ,  $Y=0$  и кривой  $Y=F(X)$ , при условии, что для всех  $X$  из отрезка  $[A, B]$  значения  $F(X) > 0$ .

*Замечание.* Процедура-функция ПЛОЩАДЬ может выполнять свое назначение для любых кривых  $Y=F(X)$ , удовлетворяющих заданным условиям. Поэтому запрограммируйте ее для кривой

$$F(X) = X^3 + 2X^2 + 1$$

(надо написать процедуру-функцию с заголовком ФУНК F(X)) на отрезке  $[0, 2]$ , разбитом на 40 частей.

Вычисленное значение площади присвойте переменной S.

## Вариант II.

Воспользовавшись алгоритмом А 11 из библиотеки алгоритмов, постройте описание процедуры-функции, вычисляющей корень уравнения  $F(X)=0$  на отрезке  $[A, B]$  с заданной точностью ЭПСИЛОН, при условии, что функция  $F(X)$  на отрезке  $[A, B]$  монотонна и меняет знак.

*Замечание* (такое же, как и в варианте I): процедура-функция УТОЧНЕНИЕ\_КОРНЯ выполняется для любых  $F(X)$ , удовлетворяющих заданным условиям. Поэтому ее надо запрограммировать, например, для  $F(X)=X^3$  на отрезке  $[-1, 1]$ .

Вычисленное значение корня присвойте переменной КОРЕНЬ.

Решение заданий самостоятельной работы

1.  
ФУНК F(X)  
НАЧ

ЗНАЧ:= $X^3 + 2X^2 + 1$

КОН

ФУНК ПЛОЩАДЬ (A, B, N)

ИМЕНА: N, I, X

НАЧ

$H := (B - A) / N$  (\*ВЕЛИЧИНА ШАГА РАЗБИЕНИЯ\*)

ЗНАЧ:=0 (\*НАЧАЛЬНОЕ ЗНАЧЕНИЕ — РЕЗУЛЬТАТУ\*)

$X := A$  (\*НАЧАЛЬНОЕ ЗНАЧЕНИЕ — ПЕРЕМЕННОЙ X\*)

$I := 1$  (\*НАЧАЛЬНОЕ ЗНАЧЕНИЕ — НОМЕРУ ШАГА\*)

ПОКА  $I \leq N$

НЦ

ЗНАЧ:=ЗНАЧ+ $H \cdot F(X)$

$X := X + H$ ;  $I := I + 1$

КЦ

КОН

S:=ПЛОЩАДЬ (0, 2, 40)

2.

ФУНК F(X)

НАЧ

ЗНАЧ:= $X^3$

КОН

ФУНК УТОЧНЕНИЕ\_КОРНЯ (A, B, ЭПСИЛОН)

ИМЕНА: C

НАЧ

ПОКА  $B - A > 2 \cdot \text{ЭПСИЛОН}$

НЦ

$C := (A + B) / 2$

ЕСЛИ  $F(A) \cdot F(C) \leq 0$

ТО  $B := C$

ИНАЧЕ  $A := C$

ВСЕ

КЦ

ЗНАЧ:=(A+B)/2

КОН

КОРЕНЬ:=УТОЧНЕНИЕ\_КОРНЯ (-1,1,0.001)

Домашнее задание: к уроку 1 — § 11, № 1, 2; к уроку 2 — № 3, 4.

указания к решению задач.

1.

ФУНК МАХ 2(M, N)

НАЧ

ЕСЛИ  $M > N$

ТО ЗНАЧ:=M

ИНАЧЕ ЗНАЧ:=N

ВСЕ

КОН

2. Одно из решений:

ФУНК МАХ 3 (M, N, S)

НАЧ

ЕСЛИ  $M > N$

ТО ЗНАЧ:=МАХ 2 (M, S)

ИНАЧЕ ЗНАЧ:= МАХ 2 (N, S)

ВСЕ

КОН

Может быть предложено и другое решение:

ФУНК МАХ 3 (M, N, S)

НАЧ

ЗНАЧ:=МАХ 2(МАХ 2 (M, N), S)

КОН

3.

ФУНК СЧЕТА (СЛОВО)

ИМЕНА:ИНДЕКС

НАЧ

ЗНАЧ:=0; ИНДЕКС:=1

ПОКА ИНДЕКС<=ДЛИН (СЛОВО)

НЦ

ЕСЛИ СЛОВО [ИНДЕКС]=«А»

ТО ЗНАЧ:=ЗНАЧ+1

ВСЕ

ИНДЕКС:=ИНДЕКС+1

КЦ

КОН

или с циклом ДЛЯ:

ФУНК СЧЕТА (СЛОВО)

ИМЕНА:ИНДЕКС

НАЧ

ЗНАЧ:=0

ДЛЯ ИНДЕКС ОТ 1 ДО ДЛИН (СЛОВО)

НЦ

ЕСЛИ СЛОВО [ИНДЕКС]=«А»

ТО ЗНАЧ:=ЗНАЧ+1

ВСЕ

КЦ

КОН

4(a).

ФУНК СЧЕТ БУКВ (СЛОВО, БУКВА)

ИМЕНА:ИНДЕКС

НАЧ

ЗНАЧ:=0; ИНДЕКС:=1

ПОКА ИНДЕКС<=ДЛИН (СЛОВО)

НЦ

ЕСЛИ СЛОВО [ИНДЕКС]=БУКВА

ТО ЗНАЧ:=ЗНАЧ+1

ВСЕ

ИНДЕКС:=ИНДЕКС+1

КЦ

КОН



4(б).

ФУНК ПРОЦ\_БУКВ (СЛОВО, БУКВА)

ИМЕНА:ИНДЕКС

НАЧ

ЗНАЧ:=0; ИНДЕКС:=1

ПОКА ИНДЕКС&lt;=ДЛИН (СЛОВО)

НЦ

ЕСЛИ СЛОВО [ИНДЕКС]=БУКВА

ТО ЗНАЧ:=ЗНАЧ+1

ВСЕ

ИНДЕКС:=ИНДЕКС+1

КЦ

ЗНАЧ:=ЗНАЧ/ДЛИН (СЛОВО)\*100

КОН

5.

ФУНК ДЕНЬ\_НЕДЕЛИ (ДЕНЬ)

ВЫБОР

ПРИ ДЕНЬ=«ПОНЕДЕЛЬНИК»:ЗНАЧ:=1

ПРИ ДЕНЬ=«ВТОРНИК»:ЗНАЧ:=2

ПРИ ДЕНЬ=«СРЕДА»:ЗНАЧ:=3

ПРИ ДЕНЬ=«ЧЕТВЕРГ»:ЗНАЧ:=4

ПРИ ДЕНЬ=«ПЯТНИЦА»:ЗНАЧ:=5

ПРИ ДЕНЬ=«СУББОТА»:ЗНАЧ:=6

ПРИ ДЕНЬ=«ВОСКРЕСЕНЬЕ»:ЗНАЧ:=7

ВСЕ

КОН

## § 12. Кортежи на Рапире (1 ч)

**Основная цель.** Познакомить учащихся со структурными данными — кортежами, расширяющими уже известное понятие табличной величины, и показать возможности представления таблиц в программе на Рапире.

**Требования к знаниям и умениям.** Учащиеся должны уметь описывать реальные списки и таблицы в виде кортежей, определять длину кортежа и элемент кортежа по его индексу, выделять из кортежа его часть и соединять кортежи.

**Методические указания.** Продолжая активно использовать аналогии с алгоритмическим языком, учитель на этом уроке сравнивает знакомые школьникам таблицы с новым понятием кортежа. Однако здесь весьма важно подчеркнуть различие этих понятий: кортеж расширяет понятие таблицы. Это подчеркивание необходимо, оно оправдывает введение нового термина для объекта, столь похожего на таблицы.

Таблицы в алгоритмическом языке представляли упорядоченную совокупность значений одинакового типа. В примерах учебника использовались главным образом таблицы, содержащие числовые значения (целые, натуральные и вещественные). В этом параграфе приводятся примеры кортежей, представляющие собой не только числовые совокупности. Например, кортеж ИМЯ\_МЕСЯЦА — это совокупность текстов, а кортеж А, сформированный по команде

$$A := \langle \langle 1, 2 \rangle, \langle 4, 4 \rangle \rangle,$$

является совокупностью двух кортежей. Гораздо важнее пример кортежа, используемого процедурами КРУГЛЫЕ\_ОТЛИЧНИКИ и ОТЛИЧНИКИ. Хотя в тексте учебника нигде явно не указана возможность включения значений разных типов в один кортеж, тем не менее в Рапире такая возможность предусмотрена, и именно она делает понятие кортежа непосредственным расширением таблиц. Учитель должен воспользоваться этим примером, чтобы продемонстрировать средства языка программирования, неизвестные в алгоритмическом языке.

В качестве элементов таблиц всегда рассматривались значения; в учебнике все примеры кортежей тоже исчерпываются кортежами с элементарными значениями — числовыми или текстовыми константами, представляющими собой частный случай выражения. В действительности же в программах на месте кортежей могут быть указаны выражения произвольного вида, вычисленные значения которых и составляют кортеж. Так, допустимо в Рапире присваивание

$$КОР := \langle 26 - 5*5, 4/2, 2*2 - 1 \rangle$$

делает значением переменной КОР кортеж  $\langle 1, 2, 3 \rangle$ , а по команде

$$ЛЕС := \langle \langle \text{«ЗА»} + \text{«ЯЦ»}, \text{«ВОЛК»}, \text{«МЕД»} + \text{«ВЕД»} + \text{«Б»} \rangle$$

переменной ЛЕС присваивается значение  $\langle \text{«ЗАЯЦ»}, \text{«ВОЛК»}, \text{«МЕДВЕДЬ»} \rangle$ .

В этих примерах сначала вычисляются значения выражений (арифметических или текстовых), а затем из этих значений формируются кортежи.

Таким образом, строгое определение кортежа (отсутствующее в учебнике) формулируется так: кортеж — это величина, представляющая собой конечную упорядоченную совокупность выражений, которые могут иметь значения произвольных типов.

Операции, которые могут выполняться с кортежами как объектами языка, во многом напоминают уже известные операции над текстами.

1. *Определение длины кортежа*, исчисляемой в количестве элементов кортежа, с помощью стандартной функции ДЛИН (Х).

В результате выполнения программы

$$ТЕК := \langle \text{«ИВАНОВ»}, \text{«ПЕТРОВ»}, \text{«СИДОРОВ»}, \text{«МИШИН»} \rangle$$

$$A := ДЛИН (ТЕК)$$

$$ЧИС := \langle 5, 4, 3, 2, 1 \rangle$$

$$B := ДЛИН (ЧИС)$$

переменной А будет присвоено целочисленное значение 4 (в кортеже ТЕК четыре элемента), а переменной В — целочисленное значение 5 (в кортеже ЧИС пять элементов).

В Рапире разрешается использовать пустой кортеж — кортеж, не содержащий ни одного элемента. Его обозначение —  $\langle \rangle$ ; длина пустого кортежа — нуль;  $ДЛИН (\langle \rangle) = 0$ .

2. *Операция выборки из кортежа* одного элемента по его номеру.

По командам

$$P := ТЕК [2]; Q := ЧИС [1]; R := ЧИС [ДЛИН (ЧИС)]$$

переменные Р, Q, R получают значения «ПЕТРОВ», 5 и 1.

Последнее значение вычисляется так:

сначала вычисляется длина кортежа ЧИС с помощью операции ДЛИН (ЧИС);

вычисленное значение (целое число 5) становится номером в операции выборки,

которая выбирает элемент кортежа ЧИС с номером 5;

выбранное значение 1 присваивается переменной R.

3. *Операция выделения фрагмента кортежа*.

Если известен кортеж А, то два целых числа N1 и N2, удовлетворяющие условиям

$$1 \leq N1 \leq N2 \leq ДЛИН (A)$$

и участвующие в операции выделения фрагмента, определяют границы выделяемого из кортежа А фрагмента, который тоже является кортежем. Например, ФРАГ:=ЧИС [2:4] делает значением переменной ФРАГ кортеж  $\langle 4, 3, 2 \rangle$ .

Будьте очень внимательны: при внешней схожести операций выборки элемента из кортежа и выделения фрагмента кортежа первую из них нельзя считать частным случаем второй. Действительно, результат операции выделения фрагмента

$$ТЕК [3:3] = \langle \text{«СИДОРОВ»} \rangle$$

представляет собой кортеж, состоящий из одного элемента, тогда как результат выборки

$$ТЕК [3] = \text{«СИДОРОВ»}$$

есть элемент кортежа — текстовая (литерная) величина.

4. *Многоуровневые выборки элементов и выделения фрагментов*.

На уроке показано, что элементами кортежа могут быть кортежи. Обычно такой объект, как кортеж кортежей, выражается довольно длинной конструкцией. Ее важно сделать выразительной, т. е. описать так, чтобы внешний вид такого описания давал представление о смысловом содержании. Задача учителя — привить ученикам навыки структурирования представления кортежей, т. е. навыки хорошего стиля программирования, например:

$$ТАБЛИЦА = \langle \langle 1, 2, 3, 4, 5 \rangle, \langle 1, 3, 5, 2, 4 \rangle, \langle 5, 4, 3, 2, 1 \rangle, \langle 2, 4, 3, 5, 1 \rangle \rangle.$$

Тогда ТАБЛИЦА [2] — это кортеж  $\langle 1, 3, 5, 2, 4 \rangle$ , а значит, к нему тоже можно применить операцию выборки элемента. Например, ТАБЛИЦА [2] [5] = 4. Как сказано в учеб-



нике, в Рапире, наряду с обозначением ТАБЛИЦА [I] [J], разрешено более привычное в математике и в алгоритмическом языке обозначение ТАБЛИЦА [I, J].

Впрочем, кортежи Рапиры предоставляют больше возможностей при описании данных, чем прямоугольные (двумерные) таблицы алгоритмического языка: количество уровней вложенности кортежей не ограничивается, а следовательно, соответствующие им «таблицы» могут иметь произвольную размерность. В Рапире может иметь смысл команда

АЛБФА:=K [2,3,2,4]

В учебнике не рассказывается о кортежах такой глубины вложенности ввиду того, что это понятие выходит за рамки прямых аналогий между таблицами и кортежами (аналогий, на которых построена методика изложения материала).

5. Операция соединения (склеивания) кортежей.

Эта операция обозначается знаком «плюс» (+). Она образует кортеж-результат из двух кортежей-«слагаемых» так, что все элементы второго кортежа (расположаемого справа от знака +) продолжают список элементов первого кортежа (расположенного слева от знака +). Следовательно, операция соединения кортежей некоммутативна: если A и B — кортежи, то, вообще говоря,  $A+B \neq B+A$ .

Учебник предлагает примеры этой операции только с кортежами-константами. Однако легко видеть, что эту операцию можно применять и к кортежам-переменным. Программа

L1:=<1,9>

L2:=<8,7>

L:=L1+L2

42 делает значением переменной L кортеж <1,9,8,7>. Таким образом, понятие выражения, которое при первом знакомстве с информатикой (алгоритмы работы с величинами, IX класс) воспринималось учащимися как известное из математики арифметическое выражение, постепенно расширяется. Сначала его расширили операции над текстами и, наконец, на этом уроке — операции над кортежами, которые могут быть как операндами выражения, так и его результирующим значением.

Учитель обращает внимание школьников на то, что кортежи и любые другие переменные могут не иметь значения, т. е. быть пустыми. Особенность пустого кортежа состоит в том, что его соединение (безразлично — слева или справа) с любым кортежем не меняет значения этого последнего кортежа. Сравните с прибавлением к числу нуля.

6. Циклы над кортежами. Цикл, последовательно обрабатывающий элементы кортежа, не является для школьников новой операцией. Однако использование циклов в работе с кортежами очень важно как в практике программирования, так и в методике освоения понятия кортежа. Поэтому на уроке следует рассмотреть три примера циклов над кортежами (процедуры КРУГЛЫЕ ОТЛИЧНИКИ, ОТЛИЧНИКИ и СРЕДНЯЯ). Хотя каждый из них может быть реализован циклом ПОКА, выбор цикла ДЛЯ здесь вполне оправдан: каждый раз переменной цикла служит номер элемента в кортеже, начальное значение этой переменной принимается равным 1, конечное значение равно длине кортежа (числу элементов), а шаг изменения переменной равен 1 (поскольку во всех таких циклах просматриваются все элементы кортежа с первого до последнего). В таких условиях цикл ПОКА сделал бы программу более громоздкой.

Домашнее задание. § 12, № 1—5, № 6 (дополнительно).

Указания к решению задач.

1.

ФУНК ПРОИЗВЕДЕНИЕ (X)

(\*ПРОИЗВЕДЕНИЕ ВСЕХ ЭЛЕМЕНТОВ КОРТЕЖА\*)

ИМЕНА:НОМЕР

НАЧ

ЗНАЧ:=1

ДЛЯ НОМЕР ОТ 1 ДО ДЛИН (X)

НЦ

ЗНАЧ:=ЗНАЧ\*X[НОМЕР]

КЦ

КОН

2. A[3]=«ТИГР»

3. ПРОЦ УСПЕВАЕМОСТЬ (= > K, СЧЕТЧИК=)

(\*В КОРТЕЖЕ К КАЖДЫЙ ЭЛЕМЕНТ ЕСТЬ КОРТЕЖ,

СОСТАВЛЕННЫЙ СВЕДЕНИЯМИ О ШКОЛЬНИКАХ:

ПЕРВЫЙ ЭЛЕМЕНТ — ФАМИЛИЯ, ВТОРОЙ — ОЦЕНКА

ПО МАТЕМАТИКЕ, ТРЕТИЙ — ПО ИНФОРМАТИКЕ\*)

ИМЕНА:НОМЕР

НАЧ

СЧЕТЧИК:=0

ДЛЯ НОМЕР ОТ 1 ДО ДЛИН(K)

НЦ

ЕСЛИ K[НОМЕР,2]<K[НОМЕР,3]

ТО СЧЕТЧИК:=СЧЕТЧИК+1

ВСЕ

КЦ

КОН

4. ПРОЦ МИН (= > K, МИНИМУМ=)

ИМЕНА:НОМЕР

НАЧ

МИНИМУМ:=K[1]

ДЛЯ НОМЕР ОТ 1 ДО ДЛИН(K)

НЦ

ЕСЛИ K[НОМЕР]<МИНИМУМ

ТО МИНИМУМ:=K[НОМЕР]

ВСЕ

КОН

МИН (ЧИСЛА,M)

5. ПРОЦ МИНОМ (= > K, МИНИМУМ=), НОМЕРМИН=)

(\*МИНИМАЛЬНЫЙ ЭЛЕМЕНТ КОРТЕЖА ПОЛУЧАЕТСЯ

В ПЕРЕМЕННОЙ МИНИМУМ, А ЕГО НОМЕР — В НОМЕРМИН\*)

ИМЕНА:НОМЕР

НАЧ

МИНИМУМ:=K[1]; НОМЕРМИН:=1

ДЛЯ НОМЕР ОТ 1 ДО ДЛИН(K)

НЦ

ЕСЛИ K[НОМЕР]<МИНИМУМ

ТО МИНИМУМ:=K[НОМЕР]

НОМЕРМИН:=НОМЕР

ВСЕ

КЦ

КОН

6. ПРОЦ ПРОЦЕНТ (= > K, ДЕВ=), МАЛ=)

(\*K — КОРТЕЖ — КЛАСС, ДЕВ — ПРОЦЕНТ ДЕВОЧЕК,

МАЛ — ПРОЦЕНТ МАЛЬЧИКОВ\*)

ИМЕНА:НОМЕР

НАЧ

ДЕВ:=0; МАЛ:=0

ДЛЯ НОМЕР ОТ 1 ДО ДЛИН(K)

НЦ

ЕСЛИ K[НОМЕР,2]=«А»

ТО ДЕВ:=ДЕВ+1

ИНАЧЕ МАЛ:=МАЛ+1

ВСЕ

ВСЕ

КЦ

ДЕВ:=(ДЕВ/ДЛИН(K))\*100

МАЛ:=(МАЛ/ДЛИН(K))\*100

КОН

7. ПРОЦ ЛАДЬЯ (= > M, = > N)

(\*M — НОМЕР ВЕРТИКАЛИ, N — НОМЕР ГОРИЗОНТАЛИ\*)

ИМЕНА:ДОСКА,НОМЕР

НАЧ

ДЛЯ НОМЕР ОТ 1 ДО 8

НЦ



```
ЕСЛИ НОМЕР/=N
  ТО ДОСКА [M,НОМЕР]:=1
```

```
ВСЕ
КЦ
ДЛЯ НОМЕР ОТ 1 ДО 8
НЦ
```

```
ЕСЛИ НОМЕР/=M
  ТО ДОСКА [НОМЕР, M]:=1
ВСЕ
```

```
КЦ
КОН
```

### § 13. Команды ввода и вывода на Рапире (1 ч)

**Основная цель.** Дать учащимся представление о командах ввода и вывода в языке программирования как выражении специфических качеств ЭВМ в роли исполнителя алгоритмов; дать понятие о диалоге пользователя с машиной.

**Требования к знаниям и умениям.** Учащиеся должны знать назначение команд ввода и вывода и уметь их правильно записывать. По тексту простой диалоговой программы они должны уметь записывать протокол общения пользователя с машиной, знать основные этапы выполнения команды ввода и уметь написать простейшую диалоговую программу.

**Методические указания.** Это единственный из посвященных языку программирования Рапира параграфов, в котором учитель не опирается на параллели между алгоритмическим языком и Рапирой. Более того, в этом параграфе излагается материал, определяющий главное отличие алгоритмического языка от Рапиры, — команды передачи информации исполнителю и от исполнителя.

Когда алгоритм, записанный на алгоритмическом языке, исполняет человек, нет проблем с передачей аргументов и результатов. Ситуация меняется, если исполнителем алгоритма становится ЭВМ. Нужно не только передать исходные данные машины, но и указать при этом, что передаваемые данные должны быть присвоены определенным переменным в программе. Вычислив некоторое значение, машина должна уметь показать его человеку — высветить на экране или напечатать на бумажной ленте печатающего устройства. Таким образом, в языке программирования, на котором записываются алгоритмы, предназначенные для исполнения на ЭВМ, оказываются необходимыми команды ввода и вывода информации.

В учебнике рассматриваются две команды ввода. Одна из них организует ввод текстов (литерных величин), другая — ввод чисел. Обе команды выполняются одинаково и отличаются друг от друга служебным словом ДАННЫХ, обязательным при вводе чисел и опускаемым при вводе текстовой информации.

По командам ввода машина выполняет следующие действия:

дойдя в исполняемой программе до команды ввода, машина останавливается и зажигает на экране символ ожидания, т. е. признак готовности воспринимать вводимую информацию; символы ожидания на разных ЭВМ (точнее, в разных системах программирования) могут оказаться различными, чаще всего применяется вопросительный знак (?);

высветив символ ожидания, машина не приступает к дальнейшему исполнению программы до тех пор, пока сидящий у машины человек не наберет на клавиатуре некоторое сообщение (текст или число); конец вводимого сообщения обозначается нажатием специальной клавиши окончания ввода (ее называют часто клавишей выполнения); на большинстве ЭВМ на этой клавише изображена «сломанная» стрелка (↵);

набранное на клавиатуре сообщение становится значением переменной, имя которой указывается в команде ввода;

после выполнения всех этих действий ЭВМ продолжает свою работу — выполняет команду, непосредственно следующую за командой ввода.

Например, команда

```
ВВОД: ОТВЕТ,
```

остановив машину, высветив символ ожидания и дождавшись реакции программиста, набравшего на клавиатуре слово ДА, присваивает текст «ДА» переменной ОТВЕТ.

Вообще говоря, ЭВМ способна самостоятельно разобраться, какого типа величину вводит с клавиатуры программист:

число всегда начинается с цифры или знака + или —;  
признак начала литерной величины — кавычка;  
первый символ в имени переменной — буква.

Однако если поручить машине различать типы величин по указанным признакам, то эти признаки надо будет задавать и при наборе вводимой информации с клавиатуры. Тогда вместо естественного ответа ДА или НЕТ на вопрос машины надо будет отвечать, записывая этот ответ вместе с окаймляющими его кавычками. Ясно, что такой способ действия существенно затруднил бы общение с машиной. Поэтому язык программирования Рапира предлагает иной способ различения типов: в зависимости от типа предусмотренной для ввода величины используются две разные команды. При этом более короткая форма команды, с единственным ключевым словом ВВОД, выбрана для ввода в машину текстовой информации. Для ввода чисел используется команда ВВОД ДАННЫХ.

Вообще говоря, команда ВВОД ДАННЫХ организует ввод не только числовых значений, но и величин всех других типов, кроме текстов. В частности, по такой команде можно ввести кортеж. Так, в ответ на символ приглашения команды ВВОД ДАННЫХ: К можно набрать на клавиатуре <5, 4, 3, 2, 1> (включая угловые скобки), и тогда значением переменной К станет кортеж <5, 4, 3, 2, 1>. В учебнике нет примеров на ввод кортежей. В методическом отношении это достаточно сложный механизм и для эффективного усвоения требует машинной практики. С другой стороны, ввод кортежа нетрудно запрограммировать в виде последовательного ввода элементов кортежа. Так, приведенную выше команду ввода кортежа К можно реализовать следующим образом:

```
ВВОД ДАННЫХ:K1; ВВОД ДАННЫХ:K2; ВВОД ДАННЫХ:K3;
```

```
ВВОД ДАННЫХ:K4; ВВОД ДАННЫХ:K5;
```

```
K:=<K1, K2, K3, K4, K5>
```

```
Или еще проще:
```

```
K:=<0, 0, 0, 0, 0>
```

```
ДЛЯ N ОТ 1 ДО ДЛИН (K)
```

```
НЦ
```

```
ВВОД ДАННЫХ:K[N]
```

```
КЦ
```

(результат будет в точности тем же, если на пять команд ввода данных на клавиатуре будут набраны соответственно значения 5, 4, 3, 2, 1).

Хотя учащийся в ходе одного урока не успеет познакомиться с дополнительными возможностями команд ввода, учителю необходимо знать некоторые из них.

Команда вывода информации более проста. Она не требует указания типа величины. Ее операндом служит произвольное выражение. Так, команда ВЫВОД:5 высвечивает на экране заданную в команде числовую константу.

Команда ВЫВОД:«СЛОНО»+«ПОТАМ» зажигает слово СЛОНОПОТАМ.

В примере, приведенном в тексте учебника, в качестве операнда команды ВЫВОД фигурирует вызов процедуры, вычисляющей функцию: ВЫВОД: НОД (А, В). Это вполне естественно, поскольку вызов процедуры-функции представляет собой частный случай выражения.

По команде ВЫВОД:2\*3.1416\*R можно выдать на экран длину окружности радиуса R, если к моменту выполнения команды вывода переменной R присвоено значение радиуса.

В этом месте урока уместен контрольный вопрос учителя классу: чем отличается выполнение двух таких программ:

```
1) ВЫВОД: 2*3.1416*R и
```

```
2) L:=3.1416*R; ВЫВОД:L?
```

О т в е т: В первой программе машина только выдает результат на экран, а вторая программа в дополнение к этому еще и запоминает результат в переменной L.

На этом уроке впервые ученикам объясняется условность записи вещественных чисел в программировании (не только в Рапире): десятичная точка вместо запятой в качестве разделителя целой и дробной части в вещественном числе. Эта условность связана с традициями программирования и прочно вошла в практику.

Команда вывода имеет и другую форму, позволяющую выводить информацию не только на экран. Так, если к ЭВМ подключено печатающее устройство — принтер, то вывод на бумагу осуществляется по команде

```
ВЫВОД НА БУМАГУ: А
```

В учебнике эта форма команды вывода не приведена.

В примерах программ, рассмотренных на уроке, команды ввода встречаются вместе с командами вывода. Это обстоятельство не случайно. Учитель должен обратить внимание учащихся на то, что, высвечивая на экране символ приглашения к вводу, машина сама по себе сообщает человеку, какую величину ему следует набрать на клавиатуре. И если в программу



включена не одна команда ввода, а несколько (как в примере процедуры НАИБОЛЬШИЙ ОБЩИЙ ДЕЛИТЕЛЬ), то символы приглашения легко могут поставить в тупик сидящего у экрана человека. Поэтому в практике программирования перед выполнением каждой команды ввода настоятельно рекомендуется написать команду вывода, которая путем вывода на экран соответствующего текста подробно (и доброжелательно) объясняет человеку, как ему надо отреагировать на очередной символ приглашения. Приведенный в учебнике разбор процедуры НАИБОЛЬШИЙ ОБЩИЙ ДЕЛИТЕЛЬ поясняет эту установку.

Перед формулированием домашнего задания учитель должен предупредить класс о предстоящей на следующем уроке контрольной работе по программированию на Рапире.

Домашнее задание. § 13, № 1—4.

Указания к решению задач.

#### 1. ПРОЦ УМОЖЕНИЕ

ИМЕНА: ПЕРВОЕ, ВТОРОЕ

НАЧ

ВЫВОД: «ВВЕДИТЕ, ПОЖАЛУЙСТА, ПЕРВОЕ ЧИСЛО»

ВВОД ДАННЫХ: ПЕРВОЕ

ВЫВОД: «ТЕПЕРЬ ВВЕДИТЕ ВТОРОЕ ЧИСЛО»

ВВОД ДАННЫХ: ВТОРОЕ

ВЫВОД: «ПРОИЗВЕДЕНИЕ РАВНО»

ВЫВОД: ПЕРВОЕ\*ВТОРОЕ

46 КОН

#### 2. ПРОЦ СОЕДИНЕНИЕ

ИМЕНА: ПЕРВЫЙ, ВТОРОЙ

НАЧ

ВЫВОД: «ПОЖАЛУЙСТА, ВВЕДИТЕ ПЕРВЫЙ ТЕКСТ»

ВВОД: ПЕРВЫЙ

ВЫВОД: «ВВЕДИТЕ ВТОРОЙ ТЕКСТ»

ВВОД: ВТОРОЙ

ВЫВОД: «РЕЗУЛЬТАТ СОЕДИНЕНИЯ ТАКОВ»

ВЫВОД: ПЕРВЫЙ+ВТОРОЙ

КОН

#### 3. ПРОЦ ПРОВЕРКА

ИМЕНА: X1, Y1, X2, Y2, ПЕРИМЕТР

НАЧ

ВЫВОД: «ВВЕДИТЕ КООРДИНАТЫ ПРЯМОУГОЛЬНИКА ABCD»

ВЫВОД: «ВВЕДИТЕ КООРДИНАТУ X ВЕРШИНЫ A»

ВВОД ДАННЫХ: X1

ВЫВОД: «ВВЕДИТЕ КООРДИНАТУ Y ВЕРШИНЫ A»

ВВОД ДАННЫХ: Y1

ВЫВОД: «ВВЕДИТЕ КООРДИНАТУ X ВЕРШИНЫ C»

ВВОД ДАННЫХ: X2

ВЫВОД: «ВВЕДИТЕ КООРДИНАТУ Y ВЕРШИНЫ C»

ВВОД ДАННЫХ: Y2

ВЫВОД: «ВЫЧИСЛИТЕ И ВВЕДИТЕ ПЕРИМЕТР ПРЯМОУГОЛЬНИКА»

ВВОД ДАННЫХ: ПЕРИМЕТР

ЕСЛИ ПЕРИМЕТР=2\*(X2-X1)+(Y2-Y1)

ТО ВЫВОД: «МОЛОДЕЦ»

ИНАЧЕ ВЫВОД: «НЕВЕРНО»

ВСЕ

КОН

#### 4. ПРОЦ НОРМА

ИМЕНА: ФАМИЛИЯ, ВЫСОТА

НАЧ

ВЫВОД: «ВВЕДИТЕ ФАМИЛИЮ СПОРТСМЕНА»

ВВОД: ФАМИЛИЯ

ВЫВОД: «ВВЕДИТЕ ВЫСОТУ»

ВВОД ДАННЫХ: ВЫСОТА

ВЫВОД: ФАМИЛИЯ

ЕСЛИ ВЫСОТА < 180

ТО ВЫВОД: «ВЫСОТА НЕ ВЗЯТА»

ИНАЧЕ ВЫВОД: «НОРМА ВЫПОЛНЕНА»

ВСЕ

КОН

#### 5. ПРОЦ РЕШЕНИЕ КВАДРАТНОГО УРАВНЕНИЯ ( $Y=>$ , $X1=>$ , $X2=>$ )

ИМЕНА: A, B, C, D

НАЧ

ВЫВОД: «КОЭФФИЦИЕНТ ПРИ КВАДРАТЕ X»

ВВОД ДАННЫХ: A

ВЫВОД: «КОЭФФИЦИЕНТ ПРИ ЛИНЕЙНОМ ЧЛЕНЕ»

ВВОД ДАННЫХ: B

ВЫВОД: «СВОБОДНЫЙ ЧЛЕН»

ВВОД ДАННЫХ: C

D:=B\*\*2-4\*A\*C

ЕСЛИ D<0

ТО Y:=«РЕШЕНИЙ НЕТ»

ИНАЧЕ Y:=«РЕШЕНИЯ ЕСТЬ»

X1:=(-B+SQRT(D))/(2\*A)

X2:=(-B-SQRT(D))/(2\*A)

ВСЕ

КОН

Язык программирования Бейсик (16 ч)

Бейсик был создан в США в 1965 г. специально как учебный язык программирования. Его название — английская аббревиатура BASIC — Beginner's All-purpose Instruction Code, что может быть переведено как «универсальный язык программирования для начинающих». Кроме того, слово basic означает «основной», «фундаментальный», «существенный». С появлением микро-ЭВМ, для многих из которых Бейсик является единственным «понятным» языком высокого уровня, он стал одним из самых распространенных в мире.

Отметим, что поскольку разные ЭВМ обладают разными возможностями, то и разработанные для них варианты (диалекты) Бейсика несколько различаются между собой. В учебнике рассказывается об основных конструкциях Бейсика, которые одинаковы почти во всех его диалектах.

Учителю следует учесть, что за 6 ч, отводимых на изучение языка программирования в школьном курсе, при краткости (или при полном отсутствии) непосредственной работы на ЭВМ возможно лишь первоначальное знакомство с Бейсиком.

При изложении Бейсика нужно подчеркнуть следующее.

1. Изменяется исполнитель алгоритмов: вместо человека — ЭВМ. В связи с этим на систему команд исполнителя накладывается новый, более высокий стандарт строгости. Повышаются требования к точности записи алгоритма. Формальный исполнитель алгоритмов — ЭВМ выполняет программы «буквально», и, например, добавление одной запятой может полностью изменить суть команды (PRINT AB приказывает печатать значение одной переменной AB, а PRINT A, B — печатать значения двух переменных A и B). На это следует обратить внимание учащихся. Если при записи алгоритмов на алгоритмическом языке допускались некоторые «вольности» (ученик мог забывать подчеркивать служебные слова, не записывать алгоритм лесенкой и т. п.), то при записи алгоритма на языке программирования от ученика следует требовать безукоризненного соблюдения синтаксических правил языка.

2. В условиях отсутствия в школе кабинета вычислительной техники учитель не может использовать наиболее эффективное средство для изучения Бейсика — ЭВМ. Учителю сложно организовать проверку программ, написанных учащимися. Здесь можно дать несколько рекомендаций:

в условиях полного отсутствия доступа к какой-либо вычислительной технике ограничиться написанием учащимися программ длиной не свыше десяти команд; при возможности «пропустить» программы, составленные учащимися, на ЭВМ длина программ может быть увеличена до 20—30 команд;

при возможности посещения дисплейного класса предприятия или другой школы необходимо организовать работу школьников на ЭВМ как в пакетном, так и в диалоговом режиме (т. е. осуществить набивку программы, ее редактирование, запуск, анализ результатов).

3. При изучении Бейсика следует опираться на знания учащихся по алгоритмическому языку.

47



Рекомендуется привить навыки перевода алгоритмов с алгоритмического языка на Бейсик.

#### § 14. Общие сведения о языке Бейсик (2 ч)

Учителю рекомендуется следующее распределение материала параграфа по урокам: 1-й урок — первое представление о Бейсике, на примерах показать, как «выглядит» программа на Бейсике; 2-й — рассказ о константах, переменных и выражениях в языке Бейсик, использовании функций в выражениях.

**Основная цель.** Дать понятие о программе на Бейсике как средстве записи алгоритма для исполнителя — ЭВМ, а также о выполнении программы машиной. Дать сведения о правилах записи выражений в Бейсике, сравнить их с соответствующими правилами алгоритмического языка. Показать, как можно использовать функции в этих выражениях.

**Требования к знаниям и умениям.** Учащиеся должны знать основные различия между алгоритмическим языком и Бейсиком, общий вид программы на Бейсике, правила записи выражений на Бейсике. Уметь записывать программу и выражения на Бейсике, а также правильно использовать функции в этих выражениях.

**Методические указания. 14.1.** При переходе от алгоритмического языка к Бейсику важно сохранить преемственность, всюду, где это уместно, подчеркнуть подобие обоих языков. Поэтому целесообразно начать с нескольких примеров рассмотренных ранее алгоритмов и параллельно дать их запись на алгоритмическом языке и Бейсике. Заголовки алгоритмов будем опускать, так как на Бейсике записывать заголовок необязательно (в первых строках программы желательно в комментариях дать общую информацию о программе для человека, а не для ЭВМ).

Пример 1 (см. пример 2 из § 7 учебника):  
на алгоритмическом языке

```
нач
цел i
S:=0
для i от 1 до 1000
нц
    S:=S+a[i]
кц
кон
```

на языке Бейсик

```
10 S=0
20 FOR I%=1 TO 1000
30 S=S+A(I%)
40 NEXT I%
50 END
```

Пример 2 (см. пример 1 из § 9 учебника):  
на алгоритмическом языке

```
нач
цел i
знач:=0
для i от 1 до длин (x)
нц
    если x[i] = «a»
        то знач:=знач+1
все
кц
кон
```

на языке Бейсик

```
10 Z%=0
20 FOR I%=1 TO LEN(XS)
30 IF MIDS(XS, I%, 1) = «a» THEN Z%=Z%+1
40 NEXT I%
50 END
```

На этих примерах уже видны главные различия между алгоритмическим языком и Бейсиком:

- 1) строки программы на Бейсике нумеруются;
- 2) вместо русских служебных слов употребляются английские:

для — FOR  
до — TO  
кц — NEXT  
кон — END  
длин — LEN  
если — IF  
то — THEN  
вырезка — MIDS

3) на алгоритмическом языке типы переменных специально указываются в заголовке алгоритма, на Бейсике же само имя переменной содержит признак типа (% для целых переменных, S — для литерных).

4) в команде присваивания вместо := употребляется обыкновенный знак равенства.

Начиная разговор о языке программирования, естественно познакомить учащихся также с тем, как заставить ЭВМ выполнить программу. В простейшем варианте человек вводит свои приказы с клавиатуры, а ЭВМ показывает результаты своей работы на экране дисплея. Пусть ЭВМ уже работает по программе интерпретатора Бейсика (см. § 17 учебника). Тогда возможны два режима ее работы: командный и программный. В командном режиме ЭВМ выполняет каждую команду отдельно, сразу же после ввода этой команды (как не программируемый микрокалькулятор). В программном режиме ЭВМ автоматически выполняет составленную заранее и введенную программу.

О готовности к работе в командном режиме ЭВМ сигнализирует выдачей на экран соответствующего сообщения, например OK (хорошо) или READY (готов). Программу можно ввести в оперативную память строку за строкой. Набрав на клавиатуре строку программы или команду, человек должен нажать клавишу ввода — только после этого информация поступит на анализ в ЭВМ. Интерпретатор отличает строку программы от команды, выполняемой сразу, по наличию номера в начале строки.

Приказ начать выполнение программы, находящейся в оперативной памяти, дается командой RUN. По ней ЭВМ переходит в программный режим и только после того, как программа выполнена, возвращается в командный режим и начинает ожидать от человека следующую команду. Рассмотрим пример отображения на экране дисплея диалога ЭВМ и человека (символы, введенные человеком, в отличие от символов, выведенных машиной, выделим курсивом).

```
OK
10X=20
20Y=100
30PRINT X+Y
RUN
120
OK
PRINT Y/X
5
OK
```

По команде PRINT (печатать) ЭВМ выводит на экран значение указанного в команде выражения.

Можно предложить учащимся выполнить следующие упражнения.



1. Знаками умножения и деления в Бейсике служат \* и / соответственно. Определите, что делает фрагмент программы

```
10 FOR I%=1 TO 100
20 KV(I%)=I%*I%
30 OBR(I%)=1/I%
40 NEXT I%
```

Ответ. Вычисляет таблицу квадратов KV и таблицу обратных величин OBR для чисел 1, 2, ..., 100. Элементам KV (0) и OBR (0) значения не присваиваются.

2. Переведите на Бейсик алгоритм копирования таблицы (см. пример 4 из § 7)

Ответ. Фрагмент программы может быть, например, следующим:

```
10 FOR I%=1 TO 100
20 B%(I%)=A%(I%)
30 NEXT I%
```

14.2. Учителю рекомендуется повторить вместе с учениками, как записывались константы, переменные и выражения в алгоритмическом языке, показать логическое сходство и синтаксические различия между записью выражений на алгоритмическом языке и на Бейсике. Одной из главных причин различий является то, что ЭВМ может обрабатывать лишь последовательности из символов, следующих друг за другом в линейной цепочке. В обычной математической нотации допустимы «многоэтажные» выражения, например:

$$\frac{a^2-1}{3x^2-15}$$

На языке программирования любое выражение должно быть «вытянуто в одну строчку».

На Бейсике предыдущее выражение запишется в виде

$(A^2-1)/(3*X^2+15)$ .

Здесь видно, что умножение в Бейсике обозначается \*, деление — /, а возведение в степень — знаком ^ . Знак умножения опускать нельзя: АВ не то же, что А\*В! Знак + используется для операций как с числами, так и с текстами.

Отделять целую часть от дробной точкой в математической нотации принято в большинстве стран мира. (Обратите внимание на то, как записываются выставленные оценки в фигурном катании, гимнастике и других видах спорта на европейских или мировых чемпионатах.) Из математической нотации это правило было перенесено в язык программирования.

Для записи больших по величине чисел и близких к нулю, надо пользоваться стандартным представлением чисел, известным из учебника алгебры для VII класса, например:

$$0.000000099 = 0.99 \cdot 10^{-7},$$

$$-451000000000 = -0.451 \cdot 10^{12},$$

и записывать эти числа в виде

0.99E-7, -0451E12.

В этой записи латинская буква «Е» сообщает ЭВМ, что для получения значения числа необходимо домножить на 10 в указанной (целой!) степени. Одно и то же число в Бейсике можно записать по-разному, например: 1000, 1E3, 0.1E4 и т. д.

Имена переменных должны начинаться с латинской буквы, а дальше можно использовать также цифры. Однако рекомендуем в именах, как и в служебных словах, употреблять (кроме цифр) только прописные латинские буквы, так как на большинстве ЭВМ строчные буквы вообще недопустимы. Имена переменных не должны содержать служебные слова языка Бейсик. Например, недопустимы имена SHIFR (содержит IF) и TON (содержит TO).

При описании таблиц в Бейсике в команде DIM указывается имя последнего элемента, а не число элементов в таблице: команда для описания таблицы А из 5 элементов правильно записывается как

```
10 DIM A (4)
```

Рассмотрим примеры, в которых знак + применяется в различных ситуациях:

```
10 A = 5
```

```
20 B = 2
```

```
30 PRINT «A+B»
```

```
40 PRINT A+B
```

```
10 AS=«МА»
```

```
20 BS+«ШИНА»
```

```
30 PRINT «AS+BS»
```

```
40 PRINT AS + BS
```

В результате выполнения этих программ на экране дисплея будет отображено соответственно

A + B

7

AS + BS

МАШИНА

В разных диалектах Бейсика используются различные наборы стандартных функций.

Как правило, в них входят:

SIN(X) — sin x

COS(X) — cos x

ABS(X) — |x|

SQR(X) —  $\sqrt{x}$

EXP(X) —  $e^x$

LOG(X) — ln x

Аргументы функций всегда должны быть заключены в круглые скобки (например, SQR (2\*X), EXP (A1+B3), SIN (BETA — GAMMA)).

Домашнее задание. К уроку 1 — § 14, с. 71, № 1—3 из дополнительных упражнений; к уроку 2 — § 14, с. 72—73, № 4—6 из дополнительных упражнений.

Дополнительные упражнения к домашнему заданию.

1. Укажите, какие из приведенных констант могут использоваться в Бейсике. Допущенные ошибки исправьте.

1) «ПРОГРАММА»

2) 0.25

3) «GAMMA»

4) -6,34

5) 0,34E-4

6) «1234»

7) 6.7.8

8) -00.24

9) A\*B+C

10) «756 4321»

11) -0.01E-2.5

12) «ПРАВИЛЬНО?»

13) «a±b»

14) МАШИНА

15) «ax<sup>2</sup>+bx+c=0»

Ответ. 4) -6.34; 5) 0.34E-4; 7) две точки; 9) это не константа, а выражение; 11) показатель степени должен быть целым; 13) нет таких символов; 14) «МАШИНА»; 15) «A\*X^2+B\*X+C=0». Остальные константы записаны правильно.

2. Запишите для использования в Бейсике следующие числа и тексты:

1) -6,789

2) 0,25×10<sup>2</sup>

3) -4,61275×10

4) ОТВЕТ

5) 3x+4y

6) -РЕШЕНИЕ-

7) 1800000000

8) 0,000014

9) σ

10) дельта

11) A1234

12) Basic

Ответ: 1) -6.789; 2) 0.25E2; 3) -4,61275E1; 4) «ОТВЕТ»; 5) «3x+4y»; 6) «-РЕШЕНИЕ-»; 7) 18E8; 8) 0.14E-4; 9) «SIGMA»; 10) «ДЕЛЬТА»; 11) «A1234»; 12) «BASIC».

3. Какие из записанных ниже последовательностей символов могут использоваться в качестве имен переменных в Бейсике? Назовите тип этих переменных (цел, вещ, лит):

1) ABC123

2) SPRINT

3) IN23%

4) IS

5) N% (10,24)

6) OPR (A)

7) ST (KLS)

8) GAMMA 1

9) Я10S

10) BETA

11) 1Z

12) %PRO

13) J

14) KLS

15) MS(1%, J%)

16) C15.6

Ответ. 1) вещ; 2) содержит PRINT; 3) цел; 4) лит; 5) цел; 6) вещ; 7) номер элемента не число; 8) в имени пробел; 9) русская буква; 10) вещ; 11) начинается с цифры; 12) начинается не с буквы; 13) вещ; 14) лит; 15) лит; 16) содержит точку.

4. Запишите по правилам Бейсика следующие выражения:

$$1) \frac{\sqrt{x^2-5}+a}{5ax}$$

$$2) 2,198x + \frac{2}{3}y$$

$$3) \frac{\sin 2x + \cos (x-y)}{\ln/x+y/}$$

$$4) -\frac{1,234+z}{z-2,895}$$

$$5) \frac{(9,87+a)(6,54+b)}{3,21c}$$

$$6) \frac{(x+1)^2}{x^2-1}$$



$$7) \left(\frac{a}{b} - e^{3a}\right)^4$$

$$8) x^{y+z}$$

$$9) e^{\sqrt{2}\sin 3x}$$

$$\text{Ответ. 1) } (SQR(X^2-5) + A) / (5 * A * X)$$

$$2) 2.198 * X + 2 / 3 * Y$$

$$3) (SIN(2 * X) + COS(X - Y)) / LOG(ABS(X + Y))$$

$$4) - (1.234 + Z) / (Z - 2.985)$$

$$5) (9.87 + A) * (6.54 + B) / 3.21 / C$$

$$6) (X + 1) ^ 2 / (X ^ 2 - 1)$$

$$7) (A/B - EXP(3 * A)) ^ 4$$

$$8) X ^ (Y + Z)$$

$$9) EXP(SQR(2 * SIN(3 * X)))$$

5. Каким из переменных A, B% или CS можно присваивать значения приведенных ниже выражений:

$$1) R(I) / 2.34$$

$$4) \text{число} + 1$$

$$7) SIN((A+B)/2)$$

$$2) (1-D\%)*(1+D\%)$$

$$5) MS + \langle ABC \rangle$$

$$8) 1E3 * X \%$$

$$3) 1.41 + 18 * X$$

$$6) (X ^ 2 - 1) / (Y ^ 2 + 2)$$

$$9) SS(7, I) ?$$

Ответ. 1) A; 2) A или B%; 3) никакой; 4) никакой; 5) CS; 6) A; 7) A; 8) A или B%; 9) CS.  
6. Запишите на Бейсике следующие команды алгоритмического языка:

$$1) a := \frac{1}{2}$$

$$2) b := \frac{3a^2}{(a+1) \left(a - \frac{1}{2}\right)}$$

$$3) c := a - b$$

$$4) d := 200000000 + c$$

$$5) x := \frac{ab(a+b)}{(c+d)cd}$$

$$6) y := 1.25x + 7.6y$$

Ответ. 1)  $A = 1/2$ ; 2)  $B = 3 * A ^ 2 / ((A + 1) * (A - 1/2))$ ; 3)  $C = A - B$ ; 4)  $D = 2E8 + C$ ;  
5)  $X = A * B * (A + B) / (C + D) / C / D$ ; 6)  $Y = 1.25 * X + 7.6 * Y$

### § 15. Команды языка Бейсик (3 ч)

Учителю рекомендуется следующее распределение материала по урокам. 1-й урок — команда ветвления; 2-й — команда повторения; 3-й — команда обращения к подпрограмме, команды ввода-вывода.

**Основная цель.** Дать учащимся сведения и отработать навыки и умения по записи алгоритмов с использованием команды ветвления (IF—THEN—ELSE), команды повторения (FOR—NEXT), команды вызова подпрограммы (GOSUB) и команд ввода-вывода.

**Требования к знаниям и умениям.** Учащиеся должны знать правила записи команд ветвления, вызова подпрограммы и ввода-вывода; уметь использовать их при записи простых программ на Бейсике, переводить простейшие алгоритмы, содержащие команды ветвления, повторения и вызова вспомогательного алгоритма, с алгоритмического языка на Бейсик.

**Методические указания. 15.1.** Следует обратить внимание учащихся на то, что команда ветвления IF—THEN—ELSE в Бейсике аналогична команде **если—то—иначе—все** в алгоритмическом языке. Учителю необходимо разобрать те синтаксические различия, которые имеются в записи команды ветвления на Бейсике и алгоритмическом языке. Прежде всего они связаны с записью серий команд после служебных слов THEN и ELSE, где может стоять серия команд, включая даже новые IF—THEN—ELSE. Команды серии разделяются двоеточием. Аналог служебного слова **все** в Бейсике не нужен — признаком конца команды ветвления служит конец строки. Пример:

100IF A>B THEN C=A:D=B ELSE C=B:D=A

По этой команде переменной C будет присвоено большее из двух чисел A и B, а D — меньшее.

В Бейсике максимальная допустимая длина строки программы (как и максимальная длина значения литерной переменной) — 255 символов. Ясно, что в одной строке на бумаге (и тем более на экране дисплея) невозможно записать столько символов. Поэтому одна «логическая» строка в принципе может состоять из нескольких «физических» строк. Начинаящим программистам советуем не увлекаться длинными строками.

Если серии команд, которые мы хотим записать после THEN и ELSE, слишком длинные и не уместятся в одной строке, то целесообразно пользоваться командой обращения к подпрограмме GOSUB (см. 15.3).

Следует обратить внимание на то, что является указателем начала и конца серий команд в командах ветвления. Первая серия начинается после слова THEN, а оканчивается либо перед словом ELSE, либо перед номером следующей строки, если слово ELSE отсутствует (сокращенная команда ветвления). Вторая серия начинается после слова ELSE, а оканчивается перед номером следующей строки. Например:

10 IF NS = «ПАРОЛЬ» THEN

PRINT «ЗДРАВСТВУЙТЕ»: F=1 1-я серия

ELSE

PRINT «ДО СВИДАНИЯ» : F=0 2-я серия

20 PRINT F

Опыт показывает, что учащиеся, хорошо усвоившие команду ветвления алгоритмического языка, без особых трудностей записывают алгоритмы, используя команду ветвления Бейсика. Учителю следует лишь серией упражнений закрепить навыки синтаксически грамотной записи команды ветвления Бейсика.

Во время урока разбирается несколько (в зависимости от подготовленности класса) примеров. Затем учащимся можно предложить самостоятельно составить короткую программу с использованием команды ветвления (например, программу нахождения числа решений уравнения  $ax + b = 0$ ). Приведенные ниже упражнения можно использовать для математического диктанта.

1. Переведите следующие отношения алгоритмического языка на Бейсик:

$$1) a > b + c$$

$$2) X \leq X^2$$

$$3) a + x = y \cdot b$$

$$4) M = \langle \text{ОТВЕТ} \rangle$$

$$5) Z = \langle \text{ОДИН} \rangle$$

$$6) x + 3.1415 \leq y - 2.28$$

$$7) z < 0.00001$$

$$8) y^2 + y \geq 5 \cdot 10^8$$

$$9) \alpha = \gamma$$

$$10) x \cdot y \cdot z = 0$$

Ответ. 1)  $A > B + C$ ; 2)  $X \leq X * X$ ; 3)  $A + X = Y * B$ ; 4)  $MS = \langle \text{ОТВЕТ} \rangle$ ; 5)  $ZS = \langle \text{ОДИН} \rangle$ ;  
6)  $X + 3.1415 \cdot Y - 2.28$ ; 7)  $Z < 1E-5$ ; 8)  $Y^2 + Y \geq 5E8$ ; 9)  $ALPHA = GAMMA$ ; 10)  $X * Y * Z = 0$

2. Определите, какие из приведенных ниже отношений Бейсика являются синтаксически неправильными, и, если можно, исправьте их:

$$1) (A > B + 3) \text{ или } (A < B)$$

$$2) A > MS(I)$$

$$3) MS + AB = TS$$

$$4) MS < > \langle \text{ВЕРНО} \rangle$$

$$\text{Ответ. 1) } (A > B + 3) \text{ OR } (A < B);$$

$$2) \text{число и текст не сравнимы;}$$

$$3) MS + \langle AB \rangle = TS$$

$$5) B < = 10; 6) A < > 3.14;$$

$$7) 0.4E-6 > X ^ 2$$

$$8) GAMMA = MS (3)$$

3. Запишите на Бейсике следующие фрагменты алгоритмов:

1) **если**  $x < A$  **и**  $x > B$   
**то**  $C := \langle B < x < A \rangle$

**все**

2) **если**  $x = x^2$



то C:=«x=0 или x=1»  
иначе C:=«x≠0 и x≠1»

все

Ответ. 1) 10 IF (X<A)AND(X>B)THEN CS=«B<X<A»

2) 10 IF X=X^2THEN CS=«X=0 или X=1»

ELSE CS=«X<>0 и X<>1»

15.2. Следует обратить внимание учащихся на то, что команда повторения FOR—NEXT в Бейсике аналогична команде повторения с параметром в алгоритмическом языке, а команда повторения WHILE—WEND в Бейсике — команде пока—нц—кц в алгоритмическом языке. Учителю необходимо разобрать синтаксические различия.

В командах повторения циклически повторяется серия команд. Команду повторения иногда называют командой цикла. Следует рассмотреть, что является указанием начала и конца повторяющейся серии команд. Серия команд начинается после заголовка цикла FOR—TO—STEP или WHILE, а заканчивается соответственно перед словом NEXT или WEND.

Отметим, что команду повторения можно записывать и в одну строчку. Это целесообразно делать, если серия повторяющихся команд состоит из 1—2 команд, например:

10 FOR L=1 TO 20:L^3:PRINT L; NEXT

Выполняя эту строку, ЭВМ отпечатает таблицу кубов чисел 1,2,...,20. После слова NEXT имя параметра цикла можно и не указывать. Необходимо разъяснить учащимся, что при любых значениях параметров цикла серия команд цикла будет выполнена хотя бы один раз.

Пример 1.

10 FOR G=8 TO 5

20 PRINT G

30 NEXT

В результате работы этой программы будет напечатано:

8

К сожалению, в большинстве версий Бейсика нет конструкции WHILE—WEND, которая удобна при переводе команды пока—нц—кц с алгоритмического языка на Бейсик. Поэтому ее следует подробно рассматривать только в школах, не имеющих возможности организовать практические занятия на ЭВМ. В остальных случаях придется отметить, что в используемой версии Бейсика эта конструкция отсутствует.

В примере на WHILE—WEND, приведенном в учебнике (с. 74), более удобно применять FOR—TO. Поэтому рассмотрим случай (пример 2), где команда повторения с параметром не помогает.

Пример 2. Пусть для данного положительного числа s требуется найти наибольшее натуральное k, такое, что  $s \geq 1^2 + 2^2 + \dots + k^2$ .

Фрагменты алгоритма и программы приведены ниже.

На алгоритмическом языке

i:=0

пока s ≥ 0

нц

i:=i+1; s:=s-i^2

кц

k:=i-1

На языке Бейсик

10 I=0

20 WHILE S>=0

30 I=I+1

40 S=S-I^2

50 WEND

60 K=I-1

В принципе конструкцию пока—нц—кц можно перевести на Бейсик и без команды WHILE, однако это требует более глубокого знания Бейсика.

На уроке желательно рассмотреть также пример вложенных циклов.

Пример 3 (см. пример 3 из § 7 учебника, с. 35).

На алгоритмическом языке

нач

цел i, j

для i от 1 до 9

нц

для j от 1 до 9

нц

произведение [i, j]:=i\*j

кц

кон

На языке Бейсик

10 DIM P% (9,9)

20 FOR I%=1 TO 9

30 FOR J%=1 TO 9

40 P% (I%, J%)=I%\*J%

50 NEXT J%

60 NEXT I%

70 END

Разобрав примеры, учитель может предложить учащимся составить самостоятельную короткую программу на Бейсике с использованием команды повторения. Материал этого урока достаточно простой. Команда повторения с параметром алгоритмического языка изучалась учащимися несколько уроков назад. Поэтому в конце урока может остаться время для повторения материала, составления простых программ, использующих команды ветвления и цикла (например, задачи на повторение 1—4 из учебника). Можно предложить учащимся выполнить дополнительные упражнения.

1. Напишите программу печати значения функции  $y=3x^2+4x-10$  для следующих значений x:

а) 0, 1, 2, 3, 4, 5

б) 10, 25, 20

в) 1,25; 1,5; 1,75; 2; 2,25; 2,5

г) -3, -2, -1, 0, 1, 2, 3.

Ответ. Для всех вариантов годится, например, одна и та же программа

20 FOR X=A TO B STEP C

30 PRINT 3\*X^2+4\*X-10

40 NEXT

к которой добавляется строка 10:

а) 10 A=0: B=5: C=1

б) 10 A=10: B=20: C=5

в) 10 A=1.25: B=2.5: C=0.25

г) 10 A=-3: B=3: C=1

от 2. Запишите на Бейсике программы построения таблиц значений следующих функций у аргумента x:

а)  $y=e^x$  для  $x=-1; -0,5; 0; 0,5; 1$

б)  $y=\ln x$  для  $x=2; 4; 6; 8; 10; 12$ .

Ответ.

а) 10 FOR X=-1 TO 1 STEP 0,5

20 PRINT EXP(X)

30 NEXT X

б) 10 FOR X=2 TO 12 STEP 2

20 PRINT LOG(X)

30 NEXT X

3. Запишите на Бейсике алгоритм из примера 4 § 8 учебника.

Ответ.

100 S=0

110 FOR I%=1 TO 100

120 S=S+A(I%)

130 NEXT I%

140 Z=S/100

150 END

4. Составьте программу для нахождения и печати наименьшего натурального N, такого что  $N^3+2N>10\,000$ .

Ответ.

10 N=1

20 WHILE N^3+2\*N<=10\,000

30 N=N+1



```

40 WEND
50 PRINT N
60 END

```

15.3. Сравнивая записи команды вызова подпрограммы на Бейсике и алгоритмическом языке, учитель на этот раз может отметить принципиальные различия. В алгоритмическом языке можно одни и те же имена переменных независимо использовать в основной программе и подпрограмме, а в Бейсике имена переменных в основной программе и подпрограмме общие.

Здесь можно напомнить алгоритм поиска большего из трех чисел (БИТ) (см. с. 46, часть 1 учебника).

Пример 1. Переведите алгоритм БИТ на Бейсик. Соответствующая программа имеет вид:

```

10 REM*БИТ — Больше Из трех *
20 INPUT A, B, C
30 GOSUB 100
40 A=C
50 B=Z
60 GOSUB 100
70 PRINT «БОЛЬШЕЕ ИЗ ТРЕХ ЧИСЕЛ», Z
80 END
100 IF A > B THEN Z=A ELSE Z=B
110 RETURN

```

Отметим, что нахождение большего из трех чисел можно записать одной командой ветвления:

```

IF A > B
THEN IF A > C THEN Z=A ELSE Z=C
ELSE IF B > C THEN Z=B ELSE Z=C

```

Уже упоминалось о целесообразности использования команды GOSUB при записи длинных серий в команде ветвления. Учителю рекомендуется разобрать пример на использование GOSUB в команде ветвления.

Пример 2. Составьте фрагмент программы, который печатает сообщение о знаке числа  $x$ , вычисляет и печатает значение функции

$$f(x) = \begin{cases} x^2, & \text{если } x > 0, \\ 0, & \text{если } x \leq 0. \end{cases}$$

Соответствующий фрагмент программы имеет вид:

```

100 IF X > 0 THEN GOSUB 200 ELSE GOSUB 300
110 PRINT TS
120 PRINT X, Y
130 END
200 TS=«X ПОЛОЖИТЕЛЬНО»
210 Y=X*X
220 RETURN
300 Y=0
310 IF X=0 THEN TS=«X=0»
    ELSE TS=«X ОТРИЦАТЕЛЬНО»
320 RETURN

```

Этот фрагмент программы отпечатает две строки. Если, например,  $X=3$ , то на дисплее будет отображено:

```

X ПОЛОЖИТЕЛЬНО
3          9
A если X=-4, то:
X ОТРИЦАТЕЛЬНО
4          0

```

Команда GOSUB является основным средством при записи алгоритмов с использованием метода последовательных уточнений. Разбор программы на этот метод послужит эффективным средством для закрепления материала по работе с подпрограммами Бейсика.

Отметим, что возвращение из подпрограммы по команде RETURN имеет варианты: в некоторых версиях, включая MSX, управление возвращается на следующую за GOSUB команду, которая иногда может находиться не на следующей, а на той же строке. Так, выполняя программу

```

100 X=2
110 GOSUB 200:X=3
120 GOSUB 200:X=4
130 GOSUB 200
140 END
200 PRINT X^X
210 RETURN

```

ЭВМ отпечатает:

```

4
27
256

```

Команды ввода-вывода необходимы для обмена информацией между ЭВМ и человеком. Исключительно важно разъяснить учащимся, что по команде PRINT ЭВМ выводит на экран значения указанных выражений (эти выражения могут быть как числовыми, так и литерными), а по команде INPUT ЭВМ дожидается от человека ввода с клавиатуры данных, которые присваиваются указанным переменным.

В школах, не располагающих вычислительной техникой, желательно в классе на доске продемонстрировать пример работы ЭВМ по программе, организующей диалог ЭВМ с человеком.

Пример 3. Составьте программу вычисления площади треугольника по заданным пользователем трем сторонам треугольника.

Соответствующая программа имеет вид:

```

10 REM*ПЛОЩАДЬ ТРЕУГОЛЬНИКА*
20 PRINT «ВВЕДИТЕ СТОРОНЫ A, B, C»
30 INPUT A, B, C
40 REM*A, B, C — СТОРОНЫ ТРЕУГОЛЬНИКА*
50 IF (A>B+C) OR (B>A+C) OR (C>A+B)
    THEN PRINT «НЕТ»:STOP
60 REM*P — ПОЛУПЕРИМЕТР*
70 P=(A+B+C)/2
80 REM*ФОРМУЛА ГЕРОНА*
90 PRINT «ПЛОЩАДЬ», SQR(P*(P-A)*(P-B)*(P-C))
100 END

```

Отметим, что строки 10, 40, 60 и 80 — это комментарии, содержащие информацию для человека (ЭВМ при выполнении программы эти строки пропускает). В более сложных программах комментарии просто необходимы, иначе очень трудно понять, что делает программа.

Пусть программа уже введена в память. Символы, вводимые человеком, выделим курсивом (чтобы отличить их от информации, выводимой ЭВМ).

Изображение на экране дисплея:

```

RUN
Введите стороны A, B, C
? 3, 4, 5
Площадь 6
Ok
RUN
Введите стороны A, B, C
? 10, 2, 5
Нет
Break in 50
Ok
RUN
Введите стороны A, B, C
? 5
?? 12, 13
Площадь 30
Ok
PRINT 2^10
1024
Ok

```

Пояснения:

Приказ выполнить команду

? указывает, что ЭВМ ждет ввода

ЭВМ ждет новых команд

Работа прервана в строке 50

?? — недостает данных

Печать в командном режиме



В конце урока учитель может предложить дополнительные упражнения:

1. Составьте программу печати таблицы значений функции  $y=10^x$  (интервал значений аргумента и шаг задает пользователь).

Ответ.

```
10 PRINT «От А до В, шаг С»
20 PRINT «Введите А, В, С»
30 INPUT A, B, C,
40 FOR X=A TO B STEP C
50 PRINT X, 10^X
60 NEXT X
70 END
```

2. Запишите на Бейсике следующие алгоритмы (значения аргументов алгоритма следует получить с помощью команды INPUT, а результат — с помощью команды PRINT):

а) алг деление (вещ  $a, b, c$ )

```
    арг а, b
    рез с
нач
    с:=a/b
```

кон

б) алг квадрат — куб (вещ  $a, c, e$ , лит  $b, d$ )

```
    арг а
    рез b, c, d, e
нач
    b:=«квадрат»
    c:=a^2
    d:=«куб»
    e:=a^3
```

кон

Ответ.

```
а) 10 INPUT A, B
    20 C=A/B
    30 PRINT C
    40 END
б) 10 INPUT A, B
    20 BS = «Квадрат»
    30 C = A^2
    40 DS = «КУБ»
    50 E = A^3
    60 PRINT BS, C
    70 PRINT DS, E
    80 END
```

Домашнее задание. К уроку 1 — § 15 (с. 73, 74), № 1 (а); к уроку 2 — § 15 (с. 74, 75), № 1 (б); к уроку 3 — § 15 (с. 76, 77), § 2—5.

Указания к решению задач.

1. Составим программу, которая проверяет значение таблицы умножения на 8. При каждом множителе I запрашивается значение произведения  $8 \cdot I$ : пока не дан правильный ответ, запрос повторяется. В конце сообщается суммарное число допущенных ошибок.

```
10 K=0
20 FOR I=1 TO 9
30 R=0
40 WHILE R <> 8*I
50 PRINT «8*», I, «=?»
60 INPUT R
70 IF R<> 8*I THEN K=K+1
80 WEND
90 NEXT I
100 PRINT K, «ОШИБОК»
110 END
```

2. На экране появятся три числа:

5 7 7

3.

```
10 REM*ПРОИЗВЕДЕНИЕ*
20 PRINT «ВВЕДИТЕ ДВА ЧИСЛА»
30 INPUT A, B
40 PRINT «ПРОИЗВЕДЕНИЕ РАВНО», A*B
50 END
```

4.

```
10 REM БОЛЬШЕЕ ИЗ ДВУХ
20 PRINT «ВВЕДИТЕ ДВА ЧИСЛА»
30 INPUT A, B
40 IF A=B THEN PRINT «ЧИСЛА РАВНЫ»
50 PRINT «БОЛЬШЕЕ ЧИСЛО»
60 IF A>B THEN PRINT A ELSE PRINT B
70 END
```

5.

```
10 REM РАЗНОСТЬ БОЛЬШЕГО И МЕНЬШЕГО
20 PRINT «ВВЕДИТЕ ДВА ЧИСЛА»
30 INPUT A, B
40 IF A>B THEN PRINT A-B ELSE PRINT B-A
50 END
```

Замечание. Строку 40 можно заменить на 40 PRINT ABS (A-B).

В заключение обратим внимание учителя на следующее. Знакомство с языком программирования нельзя ограничить разбором отдельных конструкций языка — они должны быть проработаны при решении конкретных задач. Если после изучения материала § 14 и 15 осталось время, желательно решить более сложную задачу, анализируя предложения учеников и выбирая лучшие подходы. Можно рекомендовать, например, составить программу, вычисляющую наименьшее общее кратное, или программу для ввода таблицы и нахождения в ней наибольшего и наименьшего элементов, а также среднего значения.

Задачи для классной и домашней работы следует выбирать, учитывая уровень класса. Кроме задач на повторение из учебника рекомендуем рассматривать также задачи, ориентированные на диалог ЭВМ и человека. Понимание возможностей и принципов организации такого диалога является важной составной частью компьютерной грамотности.

### Раздел III Роль ЭВМ в современном обществе.

#### Перспективы развития вычислительной техники

#### Экскурсия на вычислительный центр (ВЦ) (4 ч)

**Методические указания.** Точные рекомендации по содержанию экскурсии и методике ее проведения дать трудно. Подготавливая экскурсию, учитель должен опираться на помощь и рекомендации сотрудников ВЦ.

Можно иметь в виду следующие варианты экскурсий:

1) экскурсия проводится в школе, где имеется дисплейный класс. В этом случае целесообразно организовать практикум по решению задач на Бейсике или Рапире (интерпретаторы которых наверняка есть в составе программного обеспечения ЭВМ дисплейного класса);

2) экскурсия проводится на ВЦ предприятия или научно-исследовательского института. В этом случае необходимо:

а) продемонстрировать учащимся весь процесс решения задачи на ЭВМ (обратить внимание на работу операционной системы);

б) показать использование ЭВМ и ее прикладного программного обеспечения при решении конкретной народнохозяйственной задачи.

Разумеется, учащимся должны быть описаны технические характеристики вычислительной машины, особенности ее операционной системы, прикладного программного обеспечения, используемого в ВЦ для решения задач данного предприятия (учреждения, НИИ и т. д.). Несомненно, учащимся будет интересно поработать за дисплеем с информационно-поисковой системой или системой автоматизированного проектирования. Для того чтобы эта работа была эффективной, учитель должен вместе с сотрудниками ВЦ подготовить учебные задания, связанные с поиском конкретной информации в базе данных или вычерчиванием конкретных графических объектов. Хорошо, если после такой работы у учащихся останутся распечатки их диалога с ЭВМ, рисунки, выполненные на графопроекторе.

Можно также рекомендовать учителю обратить внимание учащихся на те вопросы,



которые, по его мнению, остались неясными при изучении разделов «Устройство ЭВМ» и «Знакомство с программированием». Так, можно наглядно продемонстрировать работу различных внешних устройств, управление станком с числовым программным управлением и т. д. Вообще любые теоретические знания, полученные учащимися при изучении курса «Основы информатики и вычислительной техники», могут быть наглядно подтверждены на экскурсии.

*Примерный план проведения экскурсии.*

Вариант 1. Дисплейный класс на основе персональных ЭВМ («Агат», ДВК-1, ДВК-2М, «Ямаха», ДЗ-28 и т. д.) или многотерминальной мини-ЭВМ (СМ-4, СМ-1420 и др.).

1. Рассказ о технических характеристиках ЭВМ, базовом программном обеспечении. Правила безопасности при работе на ЭВМ. Обучение работе с клавиатурой (1 ч).

Должны быть освещены вопросы:

основные устройства ЭВМ;

быстродействие;

объем памяти (внутренней, внешней и постоянной);

состав базового программного обеспечения.

2. Дополнительные сведения о базовом программном обеспечении. Прикладное программное обеспечение (текстовый редактор, графический редактор, учебная база данных) и работа с ним (1 ч).

На этом занятии учащиеся должны узнать о конкретных особенностях Бейсика, Рапиры или другого языка, с которым они будут работать на следующих занятиях.

3. Практикум по решению задач (2 ч).

Перечень возможных задач:

задачи на повторение № 1—10 (записать на языке программирования и исполнить на ЭВМ);

алгоритмы работы с литерными величинами (№ 5, 7, 11, § 9);

организация диалоговых программ (№ 1—5; § 13).

Вариант 2. ВЦ предприятия.

1. Задачи, решаемые ВЦ. Технические средства ВЦ (1 ч).

В ВЦ могут быть несколько ЭВМ; аналогично п. 1 варианта 1 должно быть рассказано о характеристиках каждой ЭВМ и для решения каких задач она используется. Если ВЦ оснащен дисплеями, необходимо показать школьникам, как работать с ними.

2. Базовое и прикладное программное обеспечение, служащее для решения задач ВЦ (1 ч).

Аналогично п. 2 варианта 1 на занятии учитель (или сотрудник ВЦ) рассказывает об особенностях операционной системы ЭВМ, на которой учащиеся будут работать, и о том, как пользоваться прикладным программным обеспечением. Лучше вести этот рассказ в режиме демонстрации.

3. Практикум по решению задач (2 ч).

Выше говорилось об учебных заданиях, которые учитель подготавливает совместно с сотрудниками ВЦ. Характер этих заданий зависит от того, какое программное обеспечение имеется на ВЦ. Если есть возможность поработать с текстовым редактором, можно заготовить текст с грамматическими ошибками и попросить школьников исправить их и затем распечатать текст. Работая на автоматизированном рабочем месте конструктора, школьники могут внести изменения в заранее заготовленный чертеж какой-либо детали.

Если в ВЦ нет дисплеев, можно рекомендовать следующий порядок работы (хотя это почти наверняка будет нарушением режима работы ВЦ). Школьников разбивают на группы по 2—3 человека.

Первая группа занимается подготовкой данных (набивкой на перфокарты или перфоленту). Вторая группа под руководством сотрудника ВЦ вводит в ЭВМ данные и программу. Это может быть заранее заготовленная программа, реализующая какой-либо известный школьникам алгоритм. Программа может быть написана на любом языке, транслятор с которого есть на ВЦ. Третья группа следит за ходом выполнения программы с операторского пульта, четвертая группа получает распечатку с результатами и анализирует их. Затем группы меняются местами, и процесс решения задачи (от набивки до распечатки результатов) повторяется.

Мы не даем никаких рекомендаций, если на ВЦ имеется специализированное учебное программное обеспечение. В этом случае необходимо ознакомиться с соответствующими инструкциями и методическими разработками для этого программного обеспечения.

# КАБИНЕТ ЭВТ

В настоящее время основным является безмашинный вариант курса «Основы информатики и вычислительной техники». Однако многие школы уже имеют различную ЭВТ — от программируемых микрокалькуляторов до персональных ЭВМ, и это следует учитывать.

Рекомендации Минпроса СССР предполагают использование ВТ в IX классе на факультативных занятиях в дополнение к обычным урокам; в X классе на это предусмотрены дополнительные 34 ч.

Наш журнал начинает публиковать справочно-методические пособия по использованию ВТ при изучении курса ОИВТ по пробному учебному пособию.

А. ЕРШОВ

Академик

## Решение задач с применением программируемого микрокалькулятора «Электроника БЗ-34»

Во многих школах кабинеты информатики и вычислительной техники оборудованы программируемыми микрокалькуляторами (ПМК). По сравнению с большими ЭВМ, настольными микро-ЭВМ и персональными компьютерами ПМК обладают сравнительно скромными возможностями. Тем не менее ПМК работают по тем же принципам, что и указанные машины, и могут успешно использоваться в качестве автоматических исполнителей алгоритмов разной степени сложности. Опыт программирования на машинном языке ПМК хорошо дополняет опыт использования языков программирования высокого уровня типа Рапиры или языков простых команд типа Бейсика.

В этой статье кратко описывается устройство ПМК, а также даются правила систематического перевода алгоритмов работы с числовыми величинами, записанных на алгоритмическом языке, в программы на машинном языке ПМК.

Наша промышленность выпускает несколько типов ПМК. Из них «Электроника БЗ-34» — самый распространенный, и, кроме

того, его система команд лежит в основе других ПМК. Этим объясняется выбор указанной модели.

Мы предполагаем известными правила ввода чисел в ПМК и выполнения простейших действий с ними в режиме «ручной» работы.

Как и все ЭВМ, ПМК состоит из процессора, оперативной памяти, клавиатуры и дисплея. Все эти устройства, включая блок электропитания, заключены в одном корпусе небольших размеров, уместящемся в ладони. Как и во всех ЭВМ, обрабатываемая информация и команды программы представлены внутри ПМК в двоичной системе, однако эти детали представления скрыты от пользователя и вся видимая информация имеет форму десятичных чисел и небольшого количества дополнительных знаков, высвечиваемых на дисплее.

В отличие от других ЭВМ в ПМК нет внешней памяти и принтера, а вместимость дисплея и оперативной памяти сравнительно невелика.

Оперативная память ПМК делится на чис-



ловую и командную. Числовая память состоит из 14 ячеек, имеющих адреса 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, А, В, С, Д и хранящих целые и дробные числа, содержащие до 8 значащих цифр. Командная память состоит из 98 ячеек, имеющих адреса от 00 до 97. Команды ПМК состоят только из кода операции (не имеют операндов) и кодируются двузначными либо числами, либо комбинациями цифр и дополнительных знаков. Имеется также отдельная ячейка, постоянно хранящая значение числа  $\pi$ .

Клавиатура ПМК состоит из двух переключателей (включение и выключение ПМК и переключение между радианной и градусной мерами) и тридцати клавиш, расположенных в шесть рядов. Значение клавиши обозначено знаком на самой клавише (ос-

новное значение), над клавишей (верхнее значение) и под клавишей (нижнее значение). Две клавиши имеют только основное значение, 23 — основное и верхнее, 5 клавиш нижнего ряда — основное, верхнее и нижнее значения.

Чтобы употребить клавишу в основном значении, нужно просто нажать на нее. Для употребления клавиши в верхнем значении нужно перед ней нажать на клавишу F. Нижнее значение НОП левой нижней клавиши приводится в действие предварительным нажатием на клавишу К. Нижние значения А, В, С и Д клавиш нижнего ряда приводятся в действие предварительным нажатием клавиш ИП и П.

При записи программ ПМК клавиша обозначается тем значением, в котором она употребляется в данном месте программы.

Действие клавиш F, К распространяется только на ту клавишу, которая нажимается непосредственно за ними.

Назначение отдельных клавиш клавиатуры будет поясняться по ходу изложения правил программирования для ПМК.

Процессор ПМК имеет, как и у других ЭВМ, устройство управления со счетчиком команд и арифметико-логическое устройство, содержащее 5 числовых регистров, обозначаемых X, Y, Z, T и X1.

Регистр X является главным рабочим регистром. Его содержимое постоянно высвечивается на дисплее. В него прежде всего поступает информация, набираемая на клавиатуре. Из регистра X берется аргумент для выполнения операции с одним аргументом. Для операции с двумя аргументами первый аргумент берется из регистра Y, а второй аргумент из регистра X. В регистр X направляется результат выполнения операции (с одним или двумя аргументами). Наконец, из регистра X происходит загрузка числовой памяти ПМК.

Регистры X, Y, Z и T вместе называются стеком. Это значит, что эти регистры заполняются информацией не изолированно друг от друга, а совместно. При записи или изъятии из регистра X его содержимого происходит согласованное перемещение всех четырех регистров, аналогичное нанизыванию или снятию бусинок с нити. Правила этого согласованного перемещения будут излагаться по мере знакомства с командами ПМК.

Регистр X1 используется для сохранения, в случае необходимости, содержимого регистра X.

Команды ПМК кодируются либо двузначными числами, либо двузначными комбинациями цифр и дополнительных знаков. В одной ячейке командной памяти помещается одна команда. Каждая команда име-

ет символический код, состоящий из символов тех клавиш (от одной до трех), которые нужно нажать, чтобы выполнить данную команду или ввести ее в память. Мы не будем разделять пробелами символы клавиш, если они составляют код одной команды, например

$$KX=0A$$

$$F\sqrt{\phantom{x}}$$

Дисплей. Как уже указывалось, при работе ПМК в командном режиме (см. ниже) дисплей «накоротко» соединен с регистром X и в любой момент показывает его содержимое. Дисплей содержит 12 разрядов, в каждом из которых может появиться любой из символов, изображенных на рис. 2.



Рис. 2. Символы дисплея ПМК

### Программирование конструкций алгоритмического языка

**Команда присваивания.** Для того чтобы запрограммировать на ПМК команду присваивания, ее нужно подготовить к виду, удобному для программирования. Это включает в себя следующие требования или рекомендации:

1) желательно упростить формулу, чтобы в ней не было повторного вычисления одинаковых выражений;

2) в выражении, по которому вычисляется значение величины, должны быть только те операции, которые содержатся на клавиатуре ПМК;

3) при программировании присваивания не должно случаться переполнения стека (это требование будет пояснено ниже);

4) все величины, используемые в формуле, должны к моменту выполнения программы для команды присваивания находиться в известных ячейках числовой памяти или в регистрах процессора.

Исключения могут составлять однозначные целые константы (1, 2, 3 и т. д.), которые можно помещать прямо в программе ПМК

в виде так называемых непосредственных аргументов (см. ниже).

Введем обозначение. Если K — конструкция алгоритмического языка, то K\* — программа этой конструкции на языке ПМК. Тогда способ систематического программирования команды присваивания  $Y:=E$ , где Y — переменная, а E — выражение, задается следующими правилами:

$$(1) (Y:=E) * = E * (=y) *,$$

$$(2) (=y) * = PRy,$$

где Ry — адрес числовой ячейки, сопоставленной величине y ( $Ry=0, \dots, 9, A, B, C, D$ ). При выполнении команды  $PRy$  происходит передача содержимого регистра X (в котором получается результат исполнения программы E\*) в числовую ячейку с адресом Ry.

При построении программы E\* может встретиться несколько вариантов.

(3) Если  $E=OP(e)$ , где ОП — операция с одним аргументом, то

$$E * = e * ОП *,$$

т. е. аргумент и операция меняются местами, а скобки у аргумента (если они есть) опускаются.

Так как e может быть любым выражением, то для его программирования применяются те же правила 3—5, что и для выражения E.

Операция ОП программируется подходящей командой выполнения одноаргументной операции:

$$ОП * = \begin{cases} F10X \\ F e^x \\ F lg \\ F ln \\ F arcsin \\ F arccos \\ F arctg \\ F sin \\ F cos \\ F tg \\ F \sqrt{\phantom{x}} \\ F x^2 \\ F 1/x \\ /-/- \end{cases}$$

Перед выполнением одноаргументной операции значение аргумента передается из регистра X в регистр X1 и там сохраняется.

(4) Если  $E=(e_1)ОП(e_2)$ , где ОП — операция с двумя аргументами, то

$$E * = e_1 * e_2 * ОП *,$$

т. е. оба аргумента переносятся влево от операции, а скобки, если они есть, опускаются.

Так как  $e_1$  и  $e_2$  могут быть любыми выражениями, то для их программирования применяются те же правила 3—5, что и для выражения E.

Операция ОП программируется подходя-

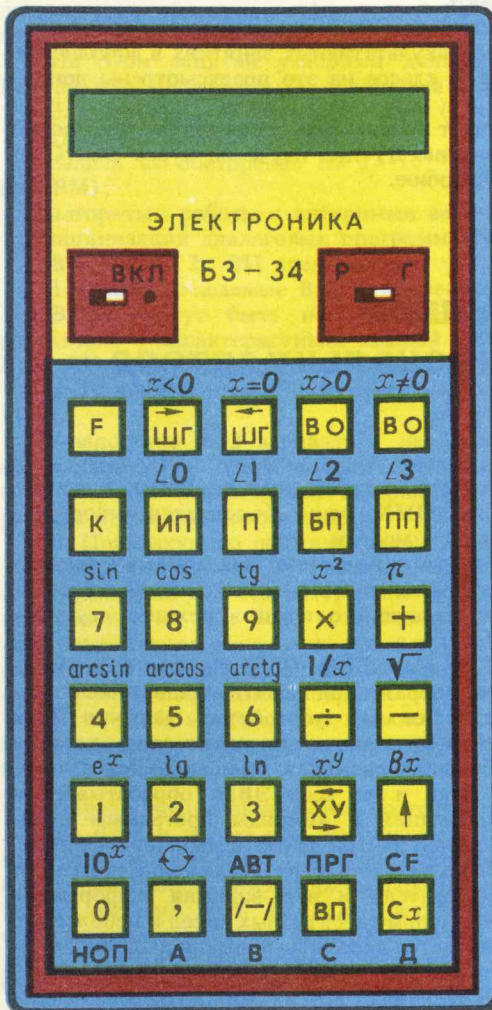


Рис. 1. ПМК «Электроника БЗ-34»



щей командой выполнения двухаргументной операции:

$$ОП^* = \begin{cases} + \\ - \\ \times \\ \div \\ FxY \end{cases}$$

Заметим, что в операции возведения  $x$  в степень  $y$  показатель степени считается первым аргументом, а основание степени  $x$  — вторым аргументом, т. е.  
 $(x^y)^* = y^*x^*Fx^y$ .

**Исключение.** Если  $E=c_1$  ОП  $c_2$ , где  $c_1$  и  $c_2$  — константы, которые вписываются в программу, то

$$E^* = c_1^* \uparrow c_2^* ОП^*.$$

Знак  $\uparrow$  — это команда подъема стека. Эта команда необходима для сохранения в регистрах ПМК обоих аргументов двухаргументных операций. Ее можно описать в виде цепочки присваиваний алгоритмического языка

$$T:=Z; Z:=Y; Y:=X$$

(старое значение  $T$  пропадает, а  $T$  принимает значение  $Z$ , которая принимает значение  $Y$ , которая принимает значение  $X$ ), т. е. все значения величин в стеке сдвигаются на одну позицию вдоль строя переменных  $X$ ,  $Y$ ,  $Z$  и  $T$ .

Принято также подъем стека изображать графически (рис. 3). Стрелки указывают передачу значений.

Смысл операции подъема стека — освободить регистр  $X$  для приема следующего аргумента двухаргументной операции.

В ПМК во всех случаях загрузки регистра  $X$  подъем стека происходит автоматически, и только в том случае, когда подряд загружаются две константы, перед второй загрузкой ставится команда  $\uparrow$ .

При выполнении двухаргументной операции первый аргумент берется из регистра  $Y$ , второй аргумент — из регистра  $X$  (сохраняясь перед этим в регистре  $X1$ ), а результат получается в регистре  $X$ . Так как аргумент в регистре  $Y$  уже «сделал свое дело», в стеке происходит спуск, т. е. обратный сдвиг значений из регистров  $T$  и  $Z$  в регистры  $Z$  и  $Y$  в соответствии с рис. 4.

**Исключение.** Спуск стека не происходит при выполнении операции  $x^y$ .

(5) Если  $E=x$ , где  $x$  — величина, которой сопоставлена ячейка с адресом  $R_x$  (0, ..., 9, A, B, C, D), то

$$E^* = ИП R_x.$$

При выполнении команды  $ИП R_x$  происходит передача содержимого ячейки  $R_x$  числовой памяти в регистр  $X$ , которая сопровождается автоматическим подъемом стека (рис. 5).

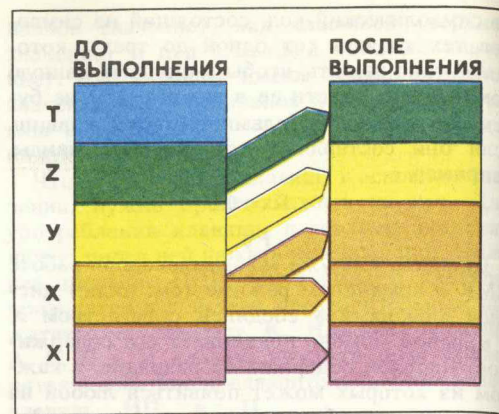


Рис. 3. Подъем стека при выполнении операции

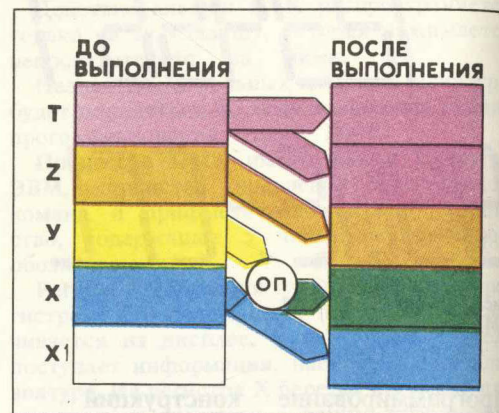


Рис. 4. Стек при выполнении двухаргументных операций

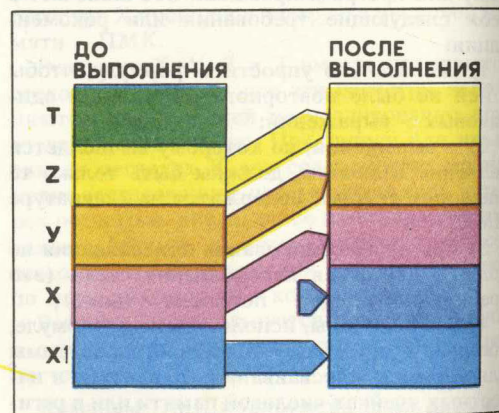


Рис. 5. Стек при взятии величины из числовой памяти

(6) если  $E=c$ , где  $c$  — константа, которая вписывается в программу, то  $E^*=c$ ,

где  $c$  — последовательность клавиш, которые нужно нажать, чтобы набрать константу на дисплее.

С использованием указанных правил удобнее всего программировать команды присваивания в следующем порядке.

1. Привести команду к виду, удобному для программирования.

2. Переписать команду присваивания по правилам 1—4, т. е. переставляя аргументы и операции и удаляя скобки, но не заменяя знаки операций и величины символами с клавиатуры ПМК.

3. Заменить знаки операции и величины символами с клавиатуры ПМК.

**Пример.**

$$y := 1/\sqrt{2\pi} e^{-x^2/2}$$

**Этап 1.** Выражение не содержит повторяющихся действий, и все операции есть в системе команд ПМК. Остается определить адреса используемых величин. Например,

$$x - 1 \\ y - A$$

Для  $\pi$  адреса не нужно, так как есть клавиша взятия этого значения  $F\pi$ . Число 2 будет записываться в программе в качестве непосредственного аргумента соответствующих операций.

**Этап 2.** При переписывании выражения удобно отмечать аргументы двухаргументных операций горизонтальными квадратными скобками:

$$2\pi \times \sqrt{\sigma 1/x} \quad \underline{2\pi \times \sqrt{1/x}} \quad \underline{x^2 \div / - / e^x x} = y$$

Получающаяся в результате этапа 2 запись выражения широко применяется в информатике и называется польской инверсной (т. е. обратной) записью, или ПОЛИЗ, в честь польского математика Яна Лукасевича, открывшего такой способ записи алгебраических выражений в 1921 г.

**Этап 3.** Обозначение величин заменяется командами передачи из памяти в регистр или командами непосредственной загрузки регистра, а знаки операций заменяются символами клавиатуры.

Получаем окончательную программу:

$$2 F\pi \times F \sqrt{F1/x} ИП1 Fx^2 \div / - / Fe^x \times PA$$

**О переполнении стека.** При описании правил приведения команды присваивания к виду, удобному для программирования, было сделано предупреждение о возможном переполнении стека. Поясним на примере, о чем идет речь.

Пусть надо запрограммировать присваивание

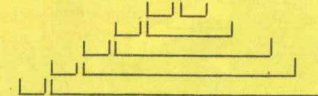
$$y := a(b+c(d-e/f)).$$

Запишем команду в одну строку и вставим знаки умножения:

$$y := a \times (b + c \times (d - e / f)).$$

Перепишем формулу в системе ПОЛИЗ, отметим горизонтальными скобками аргументы двухаргументных операций:

$$a \ b \ c \ d \ e \ f \div - \times + \times = y$$



Обозначим операции подъема стека:

$$a \uparrow b \uparrow c \uparrow d \uparrow e \uparrow f \div - \times + \times = y$$

Вспомним, что происходит со стеком  $X$ ,  $Y$ ,  $Z$ ,  $T$  при вычислении значения выражения. При каждом выполнении операции происходит подъем стека «от» регистра  $X$  (см. рис. 3), при этом значение регистра  $T$  пропадает; при каждом выполнении двухаргументной операции происходит спуск стека «в сторону» регистра  $X$  (см. рис. 4).

При выполнении указаний программы будет происходить следующее (рис. 6).

Мы видим, что после четвертого подряд подъема стека величина  $a$  добралась до регистра  $T$ , а потом исчезла, не дождаввшись начала обратного сдвига в стеке, чтобы в свое время умножиться на второй аргумент заключительной операции умножения. Такое явление и называется переполнением стека.

Для того чтобы его избежать, нужно при приведении выражения к виду, удобному для программирования, преобразовать его так,

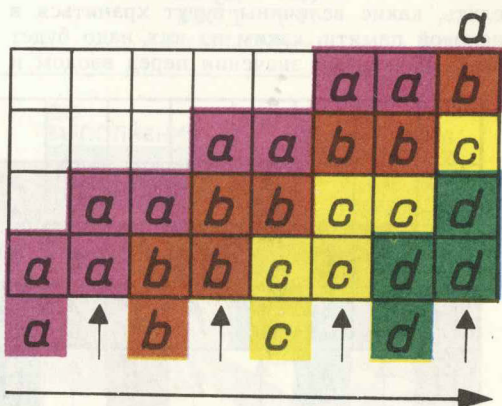


Рис. 6. Переполнение стека (нижний ряд — содержимое регистра  $X$ , второй — регистра  $Y$ , третий — регистра  $Z$ , четвертый — регистра  $T$ )

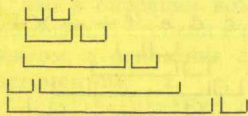


чтобы подъемы стека чередовались бы с выполнением двухаргументных операций. В случае нашей формулы это можно сделать, поменяв местами сомножители и изменив знак у внутренней скобки:

$$a(b+c(d-e/f)) = (b-(e/f-d) \times c) \times a.$$

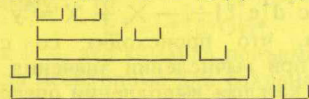
Если переписать формулу в системе ПОЛИЗ,

$$b \ e \ f / d - c \times - a \times = y$$



и обозначить операции подъема стека,

$$b \uparrow e \uparrow f \div \uparrow d - \uparrow c \times - \uparrow a \times = y,$$



66

то теперь (рис. 7) емкости стека хватит, чтобы довести вычисления до конца (результат промежуточного вычисления условно обозначается знаком заключительной операции).

**Линейный алгоритм.** Линейный алгоритм состоит из одной серии команд присваивания. Каждая из его команд программируется согласно изложенным правилам. Однако, прежде чем программировать каждое присваивание в отдельности, нужно линейный алгоритм в целом привести к виду, удобному для программирования.

Первое, что надо сделать — распорядиться постоянными и переменными величинами, используемыми в алгоритме. Во-первых, проследить, чтобы имена X, X1 и Y использовались бы только для обозначения регистров ПМК во избежание путаницы. Затем определить, какие величины будут храниться в числовой памяти, каким из них надо будет задать начальные значения перед вводом и

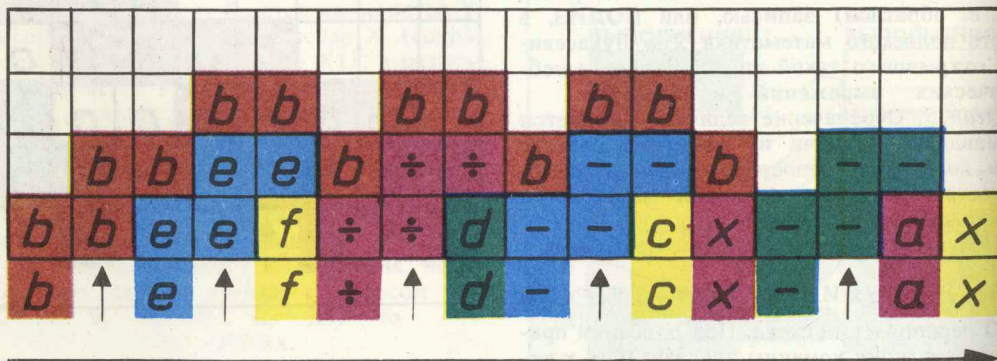


Рис. 7. Построение вычислений без переполнения стека (нижний ряд — содержимое регистра X,

запуском программы, какие будут вводиться по запросу программы в процессе ее исполнения, какие должны быть выведены и записаны в качестве результатов. Для ввода-вывода величин надо будет определить, какие из них будут направляться в память, какие только в регистр X для немедленного использования, а какие надо будет разместить в других регистрах стека. Наконец, надо предусмотреть после исполнения алгоритма остановку ПМК.

Для всех этих уточнений в запись алгоритма на алгоритмическом языке может потребоваться ввести дополнительные команды. Эти команды в основном будут иметь вид вызова вспомогательных алгоритмов (т. е. название команды, а за ним, если нужно, в скобках список параметров команды). Нам потребуются следующие дополнительные команды:

#### Команда остановки СТОП

Она завершает исполнение алгоритма.

#### Команда ввода в регистр X ВВОД

По ней исполнение алгоритма приостанавливается и предполагается, что пользователь набирает нужное число на клавиатуре, которое тем самым сразу вводится в регистр X, а потом продолжает исполнение алгоритма со следующей за вводом команды, нажимая клавишу С/П (стоп-пуск).

#### Команда ввода в числовую память ВВОД(y),

где y — обозначение переменной величины, которая по условию алгоритма размещается в числовой памяти. Эту команду пользователь исполняет так же, как и предыдущую, но в ней дополнительно обеспечивается передача числа из регистра X в требуемую ячейку числовой памяти.

второй — регистра Y, третий — регистра Z, четвертый — регистра T)

#### Команда вывода из регистра X ВЫВОД

По этой команде исполнение алгоритма приостанавливается, а на дисплее высвечивается нужный результат. После его записи пользователь продолжает исполнение алгоритма со следующей за выводом команды, нажимая клавишу С/П.

#### Команда вывода из памяти ВЫВОД(y)

Эта команда исполняется, как и предыдущая, но перед приостановкой содержимое ячейки памяти, соответствующее величине y, передается в регистр X.

Если нужно указывать дополнительные действия по сохранению введенных величин в стеке с помощью команд, имеющихся на клавиатуре, то можно использовать для них следующие названия:

СТЕК для команды  $\uparrow$ ,  
ОБМЕН (X, Y) для команды обмена содержимым регистров X и Y  
 $X:=0$  для команды очистки регистра X  
 $X:=X1$  для команды передачи из регистра X1 в регистр X FBX.

Напомним, что при этой операции в отличие от операции очистки X одновременно происходит подъем стека (рис. 8).

СДВИГ для команды F  $\odot$

При ее выполнении происходит спуск стека, но с той разницей, что содержимое регистра X не только сохраняется в X1, но и передается в регистр T, т. е. сдвиг значений происходит по кольцу (рис. 9).

Укажем теперь правила программирования команд остановки, ввода и вывода. При этом мы будем использовать звездочку для обозначения программы, соответствующей конструкции алгоритмического языка.

СТОП\* = С/П,

ВВОД\* = С/П,

ВВОД(y)\* = С/П ПРy,

где Ry — адрес ячейки числовой памяти, соответствующий величине y. Напомним, что обе команды ввода после приостановки по команде С/П требуют от пользователя ввести нужное число и продолжить исполнение программы нажатием клавиши С/П.

ВЫВОД\* = С/П,

ВЫВОД(y)\* = ИП Ry С/П,

где Ry — адрес ячейки числовой памяти, соответствующей величине y. Напомним, что обе команды вывода после приостановки по команде С/П требуют от пользователя списать выведенное на дисплей число и продолжить исполнение программы нажатием клавиши С/П.

**Пример.** Пусть надо решить систему двух линейных уравнений

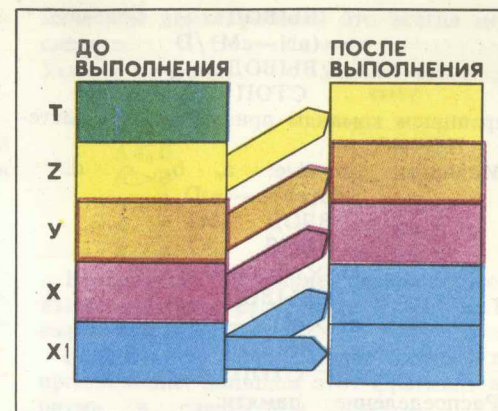


Рис. 8. Стек при выполнении команды FBX

$$ax + by = M,$$

$$cx + dy = N.$$

Для краткости ограничимся записью основной части алгоритма решения задачи:

$$D := ad - cb,$$

$$x := (Md - Nb) / D,$$

$$y := (aN - cM) / D.$$

Приведем алгоритм к виду, удобному для программирования. Предположим, что коэффициенты a, b, c и d размещаются в ячейках памяти 1, 2, 3 и 4 соответственно. Величины M и N вводятся по запросу и размещаются в ячейках 5 и 6. Промежуточная величина D помещается в ячейке 7, а корни уравнения x и y записываются сразу после их получения в регистре X.

В результате получается следующий алгоритм. Исходные данные: a, b, c, d.

$$D := ad - cb$$

$$\text{ВВОД (M)}$$

$$\text{ВВОД (N)}$$

$$x := (Md - Nb) / D$$

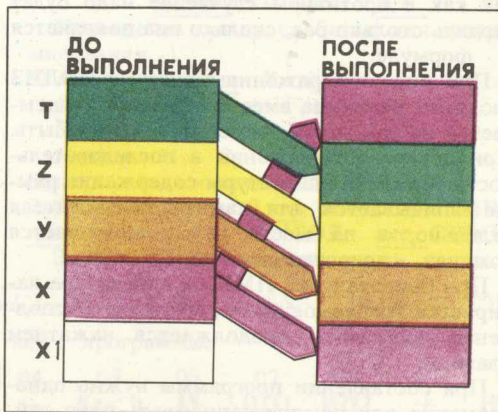


Рис. 9. Стек при выполнении команды F  $\odot$



69



**Циклы.** Циклы типа пока могут содержать те же условия с использованием регистра X, что и ветвления, т. е.

$X > 0, X < 0, X = 0, X \neq 0$

Общие правила приведения циклов алгоритмического языка к виду, удобному для программирования, тоже остаются теми же.

Покажем правила программирования цикла типа пока на примере цикла с условием  $E > 0$ , где E — выражение.

Пусть цикл имеет вид

пока  $E > 0$  нц серия кц; продолжение.

Для программирования цикла нужно знать адрес начала мп программы цикла и адрес рп начала продолжения программы вслед за командой цикла. Тогда программа цикла имеет вид

Адреса мп рп  
ячеек  
Команды  $E^* Fx > 0$  рп серия\* БП мп продолжение\*

**Пример.** Возьмем из библиотеки алгоритмов первой части курса алгоритм A6 нахождения наибольшего общего делителя:

алг НОД (нат M, N, нат НОД)

арг M, N  
рез НОД

нач нат x, y  
x := M; y := N  
пока  $x \neq y$

рых нужно найти наибольший общий делитель, вводятся пользователем по запросам, а результат выдается на дисплей.

Приведем алгоритм к виду, удобному для программирования.

Величины: x, y.

Адреса: 1, 2.

Программа располагается с ячейки 00.

ВВОД (x)

ВВОД (y)

пока  $x - y \neq 0$

нц если  $x - y > 0$

то  $x := x - y$

иначе  $y := y - x$

все

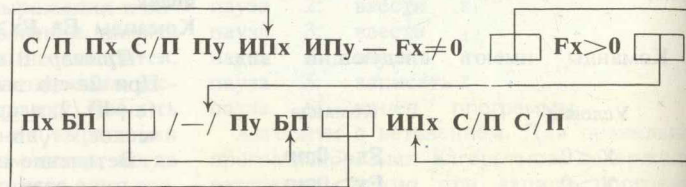
кц

ВЫВОД (x)

Выпишем команды ПМК, реализующие ветвление и цикл. Неизвестные еще адреса команд заменим стрелками. Запомним, что регистр X в процессе очередного повторения цикла хранит значение разности  $x - y$ .

ВВОД (x); ВВОД (y);  $X := x - y$ ;  $Fx \neq 0$  |  $Fx > 0$   
x := X БП |  $y := -X$  | БП | ВЫВОД (x); СТОП

Запрограммируем остальные команды, сохранив временно обозначения величин и стрелки переходов.



Запишем программу в окончательном виде.

Адреса ячеек	00	01	02	03	04	05	06	07	08	09	10
Команды	С/П	П1	С/П	П2	ИП1	ИП2	—	$Fx \neq 0$	18	$Fx > 0$	14

Адреса ячеек	11	12	13	14	15	16	17	18	19	20
Команды	П1	БП	16	—	П2	БП	04	ИП1	С/П	С/П

нц  
если  $x > y$   
то  $x := x - y$   
иначе  $y := y - x$   
все  
кц  
НОД := x  
кон

Будем считать, что числа M и N, для кото-

**Циклы с параметром.** На языке команд ПМК можно программировать простейшие циклы типа для вида

для n от N до 1 шаг — 1 нц серия кц; продолжение,

где n — натуральное число; в качестве n, называемого (обратным) счетчиком, может использоваться ячейка 0, 1, 2 или 3. Программа такого цикла имеет следующий вид:

Адреса рп  
ячеек  
Команды (n := N)\* серия\* FLn рп продолжение\*

При адресе счетчика, равном 0, 1, 2, 3, в команде повторения цикла используется клавиша L0, L1, L2 и L3 соответственно. Программа работает следующим образом.

Адреса ячеек	10	11	12	13	14	15	16	17	18	19	20
Команды	С/П	ПА	С/П	ИПА	ИПО	×	+	ПА	FL1	12	С/П

Цикл начинает выполняться при начальном значении N счетчика. После выполнения серии\* из счетчика по команде FLn вычитается единица и полученное значение сравнивается с нулем. При положительном новом значении происходит переход на команду в ячейке рп, т. е. на повторное выполнение серии\* с уменьшенным значением счетчика. Как только при очередном вычитании получится ноль, происходит переход на продолжение\*; в счетчике остается единица.

**Пример.** Возьмем в качестве примера вычисление многочлена n-й степени по схеме Горнера (алгоритм A9 из библиотеки алгоритмов первой части курса):

алг СХЕМА ГОРНЕРА (цел n, вещ x, вещ таб a [0:n], вещ y)

арг n, a, x  
рез y  
нач цел i  
i := 0  
y := A[0]  
пока i ≠ n  
нц  
i := i + 1  
y := y × x + a[i]  
кц  
кон

Приведем алгоритм к виду, удобному для программирования. Величины x и n вводятся в память перед началом исполнения программы: x — в ячейку 0 и n — в ячейку 1. Коэффициенты многочлена  $a_0, \dots, a_n$  вводятся по запросу программы в регистр X. Результат присваивается величине y, находящейся в ячейке A. Алгоритм получает вид

ВВОД (y) ввод  $a_0$  и его передача в y  
для n от n до 1 шаг — 1

Адреса ячеек	10	11	12	13	14	15	16	17	18	19	20
Коды команд	50	4—	50	6—	60	12	10	4—	5	12	50
Команды	С/П	ПА	С/П	ИПА	ИПО	×	+	ПА	FL1	12	С/П

нц ВВОД ввод очередного коэффициента в X

y := y × x + X

кц

СТОП

Выпишем команды ПМК, реализующие цикл:

ВВОД (y); ВВОД; y := y × x + X; FL1 рп; СТОП

Запрограммируем остальные команды.

С/П Пу С/П Ипу ИПх × + Пу FL1 рп С/П  
Запишем программу в окончательном виде, начиная с ячейки 10:

**Организация исполнения программы на ПМК**

**Запись программы.** Прежде чем вводить и исполнять программу на ПМК, ее нужно аккуратно записать и тщательно проверить. Записывать программу можно и в строчку, и в столбик. Запись в строчку более экономна и поэтому чаще применяется в книгах по ПМК. Так, например, выглядела бы в справочнике по ПМК программа вычисления многочлена по схеме Горнера.

**Программа вычисления многочлена**  
 $y = a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n$  по схеме Горнера.

Ввод перед исполнением программы:

x в ячейку 0, n в ячейку 1.

Ввод при исполнении программы:  
последовательность коэффициентов  $a_0, \dots, a_n$  в регистр X.

Программа (начинается с ячейки 10):  
С/П ПА С/П ИПА ИПО × + ПА FL1 12 С/П  
Результат y в ячейке A.

Однако в начале знакомства с ПМК следует записывать программы более подробно, чтобы облегчить их проверку и не ошибаться при их вводе.

Для этого полезно записывать программы в три строки: первая строка — адрес ячеек командной памяти, вторая — коды команд ПМК, третья — клавиатурные символы команд.

При вводе программы в память ПМК на дисплее для каждой вводимой команды воспроизводится ее код. Сравнение содержимого дисплея с записью команды позволяет контролировать правильность ввода программы.

Приведем пример подробной записи программы вычисления многочлена по схеме Горнера.

**Ввод программы в ПМК.** Как и другие микро-ЭВМ, ПМК имеют два режима работы: пошаговый и автоматический. В автоматическом режиме ПМК работает под управлением программы, находящейся в его командной памяти. В пошаговом режиме ПМК выпол-



няет действия, задаваемые пользователем с клавиатуры. В свою очередь, пошаговый режим делится на командный режим и режим программирования.

В командном режиме нажатие сочетания клавиш, образующего команду ПМК, вызывает ее немедленное выполнение и приостановку ПМК. В режиме программирования нажатие сочетания клавиш, образующего команду ПМК, вызывает ее ввод в очередную ячейку командной памяти, при этом на дисплее высвечивается код команды.

При включении ПМК происходит очистка всех регистров и всей памяти ПМК, а сам калькулятор оказывается в командном режиме. Переход из командного режима в режим программирования производится выполнением команды **ФПРГ**. При этом в правой части дисплея засветится адрес 00 начальной ячейки командной памяти. Это значит, что ПМК готов ввести первую команду программы в ячейку 00.

Если программу надо вводить начиная с какой-то другой ячейки **mn**, то нужно, оставаясь в командном режиме, выполнить команду безусловного перехода **БПmn**, в результате чего в счетчике команд ПМК окажется адрес **mn**, а потом перейти в режим программирования выполнением команды **ФПРГ**. В этом случае в правой части дисплея засветится адрес **mn** ячейки, в которую можно вводить первую команду программы.

При последовательном вводе программы в память значение счетчика команд в правой части дисплея увеличивается на единицу, всегда показывая адрес ячейки, «ждущей» ввода следующей команды. Одновременно в остальной части дисплея слева направо высвечиваются коды последней, предпоследней и предпредпоследней введенных команд.

Так, в ситуации, изображенной на рис. 10, следующая команда будет вводиться в ячейку 57; в ячейке 56 записана команда 10 (+), 55 — команда 1 E (Ftg), 54 — команда ввода числа 2 в регистр X.

Если при вводе программы командная память уже чем-то загружена, то при вводе очередной команды в некоторую ячейку памяти ее старое содержимое пропадает.

В режиме программирования можно не только вводить программу, но и читать ее, перемещая точку чтения и записи с помощью клавиш **ШГ** и **ШГ**. При нажатии клавиши **ШГ** значение счетчика команд, указанное в



Рис. 10. Пример индикации при вводе программы

правой части дисплея, увеличивается на единицу, а при нажатии **ШГ** — уменьшается на единицу. При этом в любой момент на дисплее показываются последние три команды, предшествующие текущему значению счетчика команд.

С помощью клавиш **ШГ** и **ШГ** можно просматривать введенную программу и исправлять ошибки ввода. Если при исправлении программы нужно в какой-то ячейке стереть находящуюся там команду, для этого достаточно ввести в эту ячейку команду **КНОП**, означающую «нет операции».

**Исполнение программы.** Готовность ПМК к автоматическому режиму задается командой **ФАВТ**. Эту команду можно выполнять и в командном режиме, и в режиме программирования. После выполнения этой команды можно продолжать «ручную» работу на ПМК, выполняя отдельные команды, но при первом же нажатии на клавишу **С/П** калькулятор начнет выполнять программу начиная с текущего значения счетчика команд. Таким образом, если надо выполнить программу с ячейки 00, достаточно произвести следующие действия:

**ФАВТ В/О** (команда установки нуля в счетчик команд) **С/П**

Если нужно начать выполнение с некоторой ячейки **mn**, то нужно перед пуском выполнить команду безусловного перехода

**ФАВТ БП mn С/П**

При остановке исполнения программы по команде **С/П** калькулятор остается в состоянии готовности к автоматическому режиму.

Исполнение программы обычно сопровождается дополнительными действиями пользователя по вводу данных и чтению результатов. Все эти действия, вместе взятые, должны в определенном порядке и без ошибки выполняться перед исполнением машинной программы или в паузах при ее приостановках. Для того чтобы не ошибиться при этом диалоге с ПМК, пользователь должен заранее записать эту программу для себя.

Вот как может выглядеть такая программа пользователя при исполнении программы схемы Горнера для вычисления многочлена  $62,135x^4 - 12,721x^3 + 5,965x^2 + 123,156$  при  $x = 7,121$ .

Включить калькулятор

**БП 10**

**ФПРГ**

Ввести программу

**ФАВТ**

**4 П1** ввод  $p=4$  в ячейку 1

**7,121 ПО** ввод  $x$  в ячейку 0

**БП 10**

**С/П**

**62,135** ввод  $a_0$

**С/П**  
**12,721** ввод  $a_1$   
**С/П**  
**5,965** ввод  $a_2$   
**С/П**  
**0** ввод  $a_3$

Е. ИКАУНИЕКС

## Основы языка Бейсик

Эту статью рекомендуем учителю использовать (после изучения § 14 и 15 учебника и соответствующих параграфов методического пособия) в качестве углубленного, систематического курса языка Бейсик, где можно почерпнуть дополнительную информацию и задачи для уроков, а также в качестве справочника по конструкциям Бейсика.

Среди многочисленных версий здесь выбран Бейсик, которым оснащены школьные КУВТ «Ямаха» и на котором разрабатывается программное обеспечение школьного курса информатики.

Рассматриваются только основные конструкции языка (полностью опущены, например, команды графики и звука). Отличия в версиях Бейсика для микро-ЭВМ ДВК-2М и мини-ЭВМ СМ-4 приведены в п. 18. Надеемся, что этого материала будет достаточно для обучения составлению и отладке простых программ.

### 1. Сведения о MSX Бейсике

Каждый язык программирования имеет свои грамматические правила. Первое из них гласит: в записи программы можно использовать только символы из алфавита языка.

Алфавиты различных языков программирования мало отличаются друг от друга; они содержат все символы, представленные на клавиатуре соответствующей ЭВМ.

Алфавит Бейсика состоит из:

- 1) прописных латинских букв  
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
- 2) цифр  
0 1 2 3 4 5 6 7 8 9
- 3) специальных символов  
. точка  
, запятая  
: двоеточие  
( точка с запятой  
( левая скобка  
) правая скобка

**С/П**  
**123,156** ввод  $a_4$   
**С/П**  
**ИПА**  
Записать ответ 155 603,98.

- пробел
- " кавычка
- ' апостроф
- ! восклицательный знак
- ? вопросительный знак
- + плюс
- минус
- \* звездочка (знак умножения)
- / знак деления
- ^ знак возведения в степень
- = равно, знак присваивания
- > больше
- < меньше
- \$ валютный знак
- α коммерческое по
- & коммерческое и
- % процент
- # номер
- \ обратная косая черта
- \_ символ подчеркивания
- 4) дополнительных символов:
  - a) строчных латинских букв  
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

- б) прописных русских букв  
А Б В 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 Э Ю Я
- в) строчных русских букв  
а б в г д е ж з и к л м н о п р с т у ф х ц ч ш щ ъ ы э ю я

Цифру 0 легко перепутать с буквой О, поэтому в записях программ принято нуль перечеркивать. Типографские же символы для нуля и буквы О различны, и в пособии перечеркнутый нуль использоваться не будет.

Пробел — такой же символ алфавита, как все другие, хотя в позиции, занятой им, на экране дисплея и в распечатках никакой графический символ не изображается (просто она остается пустой). Обозначение — применяется только при записи программ на бумаге, если хотят пробел сделать «видимым» (например, если важно точное число пробелов).

Все, что мы хотим сказать на Бейсике, необходимо кодировать буквами, цифрами и специальными символами. Дополнительные



74

3. Какие значения может принимать индекс элемента массива? (Целые неотрицательные от 0 до максимального значения, указанного в объявлении.)



#### Упражнения.

1. Какие переменные соответствуют именам A; C6\$; K7\*; N %; M %\$; IW; F8; L5 %; D\$?

Ответ: вещественная; литерная; недопустимое имя; целая; недопустимое имя; недопустимое имя; вещественная; целая; литерная.

2. Сколько элементов в массивах, объявленных командой DIM P4(6); Z(99,3); S\$(20)?

Ответ: 7; 400; 21.

3. Написать объявления массивов:

а) V(0), V(1),..., V(22);  
б) N % (0,0), N % (0,1),..., N % (7,12).  
Ответ: а) DIM V(22); б) DIM N % (7,12).

#### 4. Числовые выражения и стандартные функции

В отличие от алгоритмического языка, где алгебраические выражения записывались в обычной, принятой в математике, форме, в языках программирования необходима большая строгость. Дело в том, что в обычной математике запись иногда может быть понята по-разному (что такое  $\lg x$ : логарифм  $x$  или произведение трех величин  $l, g, x$ ?), часто используются записи, размещающиеся

не в строку (например,  $\frac{a^2}{3}$ ) и т. п.

Для обозначения арифметических операций в Бейсике применяются символы + (сложение), - (вычитание), \* (умножение), / (деление) и ^ (возведение в степень). Используя их, а также числа (постоянные) и имена переменных, можно образовывать выражения так же, как это дела-

ется в алгебре. Знак умножения нельзя пропускать!

Если порядок выполнения операций специально не задан с помощью скобок (правила употребления скобок те же, что и в алгебре), то ЭВМ в первую очередь выполнит возведения в степень, потом умножения и деления, и в последнюю очередь — сложения и вычитания; операции с одинаковыми приоритетами выполняются в порядке их прочтения — слева направо.

Пример.

Обычная запись	Запись на Бейсике
$x^5 - 4ab(x+y)$	$X^5 - 4 * A * B * (X + Y)$
$\frac{1}{\frac{1}{A} - \frac{1}{B}}$	$1 / (1 / A - 1 / B)$
$\frac{Ck+1}{ab}$	$C ^ (K + 1) / (A * B)$ или $C ^ (K + 1) / A / B$

Те элементарные функции, сокращенные имена которых «понимает» интерпретатор Бейсика, будем называть стандартными функциями. Их использование позволяет записать на Бейсике все встречающиеся в элементарной математике выражения. Постоянные и имена переменных — это самые простые частные случаи выражений.

Примеры.

$\sin 2x - \lg x$	$SIN(2 * X) - TAN(X)$
$\ln  a+b $	$LOG(ABS(A+B))$
$\sqrt{\arctg 4x}$	$SQR(ATN(4 * X))$
$e^{x^2}$	$EXP(X^2)$

Приведем наиболее часто используемые стандартные функции (аргумент за именем функции обязательно заключается в скобки).

$\sqrt{x}$	SQR(X)	(SQuare Root — квадратный корень)
$e^x$	EXP(X)	(EXPOntial function — показательная функция)
$\ln x$	LOG(X)	(natuRal LOGarithm — натуральный логарифм)
$\sin x$	SIN(X)	(SINe — синус)
$\cos x$	COS(X)	(COSine — косинус)
$\lg x$	TAN(X)	(TANgent — тангенс)
$\arctg x$	ATN(X)	(ArcTaNgent — арктангенс)
$ x $	ABS(X)	(ABSolute value — абсолютное значение)
$[x]$	INT(X)	(INTeger part — целая часть)
$\text{sig } x = \begin{cases} 1, & \text{если } x > 0, \\ 0, & \text{если } x = 0 \\ -1, & \text{если } x < 0 \end{cases}$	SGN(X)	(SiGN — знак)
	RND(X)	(RaNDom number — случайное число)

Аргумент тригонометрических функций и значение функции ATN(X) выражаются в радианах.

INT(X) округляет аргумент до ближайшего целого числа, не превосходящего аргумент: INT(3)=3, INT(4.9)=4, INT(-5.2)=-6.

Во многих версиях Бейсика нет стандартной функции для десятичного логарифма, и приходится выражать  $\lg x$  через  $\ln x$ :

$$\lg x = LOG(X) / LOG(10).$$

Аналогично надо поступать, если необходимо записать прямые и обратные тригонометрические функции, не представленные

в списке стандартных функций, или вычислить тригонометрические функции при аргументах, выраженных в градусах:

$$\arcsin x = ATN(X / SQR(1 - X * X)),$$

$$\sin x_{\text{град}} = SIN(X_{\text{рад}} * 3.14159265 / 180).$$

С помощью функции RND(X) можно получать случайные числа в границах от 0 до 1. Если аргумент положителен, то ЭВМ вычисляет случайное число, используя предыдущее вычисленное ею же случайное число; если аргумент отрицателен, то значение RND(X) зависит только от X, для различных X получаются различные случайные числа; RND(0) просто повторяет предыдущее случайное число. Стандартная функция RND(X) необходима при моделировании случайных процессов (например, в играх). Так, если мы хотим, чтобы ЭВМ «бросала» игральную кость и выдавала одно из равновероятных значений 1, 2, 3, 4, 5, 6, то можно использовать выражение

$$INT(RND(1) * 6) + 1.$$

В выражениях можно свободно использовать также элементы массивов с индексами-выражениями, если только их значения не выходят за пределы, заданные при объявлении соответствующего массива,

$$M1(INT(LOG(1 + C * (1 + G))))).$$

Вопросы.

1. В чем главные различия между записью выражений на Бейсике и в математике? (В Бейсике все выражение должно быть записано в строчку, можно применять только допустимые имена переменных, знак умножения \* нельзя опускать.)

2. В какой последовательности выполняются арифметические операции? (Сначала вычисляются значения выражений, заключенных в скобки, а из операций в первую очередь выполняется возведение в степень, во вторую — умножение и деление, в третью — сложение и вычитание.)

3. Какие стандартные числовые функции есть в Бейсике? Опишите их.

Упражнения.

1. Записать на Бейсике выражения:

$$а) \frac{(ax^2 + vx + c)^3 + 1}{(a+b)^x};$$

$$б) \frac{\sin^2 x}{\lg x + \cos 2x};$$

$$в) \lg(|4 + \sqrt{1-x}| - 2x);$$

$$г) 2 \sin 1^\circ 30'.$$

Ответ.

$$\begin{aligned} & ((A * X^2 + 2 * B * X + C)^3 + 1) / (A + B * X); \\ & (SIN(X))^2 * (TAN(X) + COS(2 * X)); \\ & LOG(ABS(4 + (1 - X) * 1(4))) - \\ & - 2 * X / LOG(10); \\ & 2 * SIN(1.5 * 3.14159265 / 180). \end{aligned}$$

2. Написать на Бейсике выражение для большего корня уравнения

$$ax^2 + vx + c = 0 (v^2 - 4ac > 0).$$

Ответ.

$$(-B + SQR(B * B - 4 * A * C)) / (2 * A).$$

3. Написать на Бейсике общую формулу для величины угла (в градусах), тангенс которого равен у.

Ответ.  $180 * (ATN(Y) / 3.14159256 + N).$

4. Написать выражение, пригодное для моделирования бросания монеты, т. е. выдающее 0 или 1 (с равными шансами).

Ответ.  $INT(2 * RND(1)).$

#### 5. Литерные выражения и стандартные функции

Рассмотрим теперь средства обработки текстов. Очень часто приходится составлять текст, присоединяя в конце одной символьной строки другую. Эта операция называется соединением и обозначается, как и в алгоритмическом языке, знаком +. Так, значением выражения «НЕ» + «ЖЕН» является строка «НЕЖЕН». Отметим, что вторая символьная строка приписывается сразу за первой, без вставления пробелов.

Длина текста вычисляется стандартной функцией LEN(X\$) (LENGth — длина). На алгоритмическом языке эта функция обозначалась через длин. Аргументом может служить любое литерное выражение, например, LEN(«С» + «ПОРТ») имеет значение 5.

Вырезку из текста реализует стандартная функция MID\$(X\$, M, N) (MIDdle — середина), дающая часть символьной строки X\$, длиной в N символов, начиная с M-го символа (в алгоритмическом языке ей соответствует выражение  $X[M:M+N-1]$ ), например, MID\$(«КЛАВИАТУРА», 3, 4) = «АВИА», MID\$(«генерал» + «литр», 5, 5) = «ралли».

Первым аргументом этой функции может служить любое символьное выражение, двумя другими — любые числовые выражения с целыми значениями.

Первые M символов символьной строки X\$ позволяет получить стандартная функция LEFT\$(X\$, M) (LEFT — левый), например, LEFT\$(«деньги», 4) = «день».

Аналогично, последние M символов строки дает функция RIGHT\$(X\$, M) (RIGHT — правый), например, RIGHT\$(«гамбит», 3) = «бит».

Весьма полезны также функции преобразования литерной строки в число и наоборот. VAL(X\$) (VALue — значение) дает числовое значение символьной строки X\$, рассматриваемой как запись десятичного числа: VAL(«1E3») = 1000, VAL(«-0.8») = -0.8.



Если начало символьной строки не содержит числа, то значение функции приравнивается нулю,  $VAL(«A1») = 0$ ,  $VAL(«1A») = 1$ .

Обратная к ней — функция  $STR$(X)$  (STRing) — строка) дает по значению числа X его запись в виде символьной строки:  $STR$(1E2) = «100»$ ,  $STR$(-1/3) = «-.3333333333333333»$ .

Функция  $CHR$(X)$  (CHaRacter — символ) выдает символ с кодом X,  $0 \leq X \leq 255$ . Например,  $CHR$(34)$  выдает символ — кавычку, которую в такой форме можно включить в символьную строку; значением выражения  $CHR$(34) + «Agat» + CHR$(34)$  является символьная строка из 6 символов: «, A, г, а, т, ».

Обратная функция  $ASC(X$)$  (ASCII — Американский стандартный код обмена информацией) по символу X\$ выдает его код; например,  $ASC(« ») = 32$ .

78

Набор всех этих стандартных функций позволяет образовывать любые литерные выражения, необходимые для обработки текстовой информации.

#### Вопросы.

1. Какие стандартные функции имеют своим значением символьные строки? ( $MID$(X$, M, N)$ ,  $LEFT$(X$, M)$ ,  $RIGHT$(X$, M)$ ,  $STR$(X)$ ,  $CHR$(X)$ .)

2. Какие стандартные функции с аргументами-символьными строками имеют числовые значения? ( $LEN(X$)$ ,  $VAL(X$)$ ,  $ASC(X$)$ .)

#### Упражнения.

1. Пусть  $A$ = «теленки»$ ,  $B$ = «около»$ ,  $C$ = «программы»; чему равны значения выражений:$

- $LEFT$(A$, 4) + RIGHT$(C$, 6)$ ;
  - $RIGHT$(B$, 4) + MID$(B$, 2, 3)$ ;
  - $«E» + MID$(A$, 3, 3) + «a»$ ?
- Ответ. а) «телеграммы»; б) «колокол»;

в) «Елена».

2. Чему равно  $VAL(STR$(5))$ ?

Ответ. 5.

3. Чему равно,  $ASC(CHR$(Y))$ ?

Ответ. Y, если  $0 \leq Y \leq 255$ .

4. Известно, что  $ASC(«0») = 48$ ,  $ASC(«1») = 49$ , ...,  $ASC(«9») = 59$ . Чему равно  $VAL(CHR$(50) + CHR$(55) + CHR$(52))$ ?

Ответ. 274.

#### 6. Команда присваивания

В Бейсике общий вид команды присваивания

$LET \langle \text{имя переменной} \rangle = \langle \text{выражение} \rangle$  ( $LET$  — пусть). Ключевое слово  $LET$  можно опускать. Например,

$LET X = A + B$ ,  $M$(1) =  $MID$(X$, M, K)$ ,  $Q3 = Q3 + 3$ .$

В команде присваивания типы переменной и выражения должны быть согласованы: нельзя числовой переменной присваивать литерное значение и наоборот.

Перед тем, как программе понадобится значение переменной, оно должно быть присвоено. Однако в случае, когда программист об этом не позаботился, интерпретатор Бейсика сам автоматически присвоит «дежурные» начальные значения: 0 — числовым переменным и пустое слово «» — литерным.

#### Вопросы.

1. Чем отличаются записи команды присваивания в Бейсике и в алгоритмическом языке? (В Бейсике употребляется знак присваивания  $=$ , а в алгоритмическом языке знак  $:=$ .)

2. Каковы ограничения на переменную и выражение, входящее в команду присваивания? (Они должны быть одного типа).

3. Что происходит со значениями переменных, имена которых встречаются в правой части команды присваивания? (Ничего; они только используются при вычислении значения выражения.)

#### Упражнения.

1. Какие значения последовательно получает при выполнении серии команд

$X = 5$   
 $X = X^3 - X^2$   
 $X = SQR(X) - 3$   
 $X = (X - 1) / 2$

переменная X?

Ответ. 5, 100, 7, 3.

2. Какие значения получат после выполнения серии команд

$I = 1$   
 $M(I) = I + 1$   
 $M(M(I) - 2) = 1$   
 $M(1) = M(0) - M(1)$

элементы массива  $M(0)$  и  $M(1)$ ?

Ответ.  $M(0) = 1$ ,  $M(1) = 3$ .

Каков результат выполнения следующих команд:

а)  $Y$ = X + 7$ ;  
б)  $X = 3$   
 $Y = 1 / (X * X - 2 * X - 3)$ ?

Ответ. а) ошибка — литерной переменной нельзя присвоить числовое значение; б) ошибка — вторая команда требует деления на 0.

#### 7. Команда вывода информации

Бейсик позволяет просто реализовать диалог пользователя и ЭВМ. Суть диалога состоит в том, что в процессе автоматического

выполнения программы (приостанавливая на время это выполнение) ЭВМ может «попросить» пользователя ввести дополнительные данные или ответить на вопрос. Дальнейшая работа программы и ее результаты будут зависеть от ответов пользователя.

Программист может организовать «разговор» пользователя и ЭВМ с помощью команд  $PRINT$  (печатать) и  $INPUT$  (ввести). Рассмотрим первую из них.

По команде  $PRINT$  ЭВМ может сообщить о результатах своей работы, показывая на экране дисплея вычисленные значения выражений или подготовленные тексты. Простейший вариант команды —

$PRINT \langle \text{выражение} \rangle$

Выполняя ее, ЭВМ вычислит значение данного выражения и покажет его на экране. Так, в результате выполнения серии команд

$X = 3$   
 $PRINT 2 * X + 5$

ЭВМ выведет число 11. В частном случае выражение может быть просто литерной постоянной: выполнение команды

$PRINT \langle \text{ПРИВЕТ!} \rangle$

состоит в показе на экране слова ПРИВЕТ!

В одной команде  $PRINT$  можно потребовать вывода значений нескольких выражений; они будут размещаться на одной и той же строке экрана.

Число позиций в одной строке зависит от типа ЭВМ. На некоторых ЭВМ можно выбрать шрифты разной величины — тем самым меняется и число символов в строке. Впредь для определенности будем считать, что в строке помещается 40 символов, позиции нумеруются, начиная с нуля. Строка считается разбитой на зоны; будем считать, что длина зоны 14 символов, т. е. первая зона занимает позиции с 0-ой по 13-ю, вторая с 14-й по 27-ю и т. д.

Если в одной команде  $PRINT$  записывается несколько выражений, их следует отделять друг от друга точкой с запятой или просто запятой. В первом случае значение текущего выражения выводится сразу за значением предыдущего выражения. Во втором случае (,) вывод продолжается с начала следующей зоны. Например, выполнение серии команд

$X = -5$   
 $PRINT \langle \text{SGN}(X; X; X) = \rangle; \text{SGN}(X)$

приведет к выводу строки

$\text{SGN}(-5) = -1$

Команда

$PRINT \langle \text{ПРИВЕТ!}, 2 * 2 = 4 \rangle$

выведет строку

ПРИВЕТ! 2\*2=4

Каждая новая команда  $PRINT$  начинается вывод с новой строки. Только в случае, если предыдущая команда  $PRINT$  кончилась точ-

кой с запятой, следующая команда  $PRINT$  продолжит вывод в уже начатой строке, например,

$PRINT \langle A \rangle$ ;  
 $PRINT \langle B \rangle$   
 $PRINT \langle C \rangle$

приведет к выводу двух строк

AB

C

Команда

$PRINT$

выведет пустое слово, т. е. просто обеспечит переход на следующую строку.

Если мы хотим начать текст не с начала строки, можно воспользоваться стандартной функцией  $TAB(X)$ . Например,

$PRINT TAB(5) \langle A \rangle$ ;  $TAB(10) \langle B \rangle$

выведет строку:

A B

Разумеется, тот же эффект дает команда

$PRINT \langle \quad A \quad B \rangle$

Вопросы.

1. Зачем нужен диалог ЭВМ и пользователя? (Чтобы пользователь мог в ходе выполнения программы влиять на дальнейшие вычисления.)

2. Как добиться, чтобы ЭВМ сообщила пользователю результаты своих вычислений? (В программу надо включить соответствующую команду  $PRINT$ .)

#### Упражнения.

1. Что появится на экране в результате выполнения серии команд

$A = 8$   
 $B$ = «BASIC»$   
 $PRINT B$, B$$   
 $PRINT$   
 $PRINT TAB(A) B$; «—ЭТО ЗДОРОВО!»$

Ответ.

BASIC BASIC — ЭТО ЗДОРОВО!

2. Написать серию команд, выдающую на экран:

а) ПРОГРАММИРОВАНИЕ ПРОГРАММИРОВАНИЕ ПРОГРАММИРОВАНИЕ  
б) таблицу значений i и x' при  $i = 1, 2, 3$ .

Ответ.

а)  $P$ = «ПРОГРАММИРОВАНИЕ»$

$PRINT P$ + « » + P$ + « » + P$$

б)  $PRINT 1, X$

$PRINT 2, X * X$

$PRINT 3, X^3$

(Конечно, возможны и другие варианты решений.)

#### 8. Режимы работы ЭВМ

Языки программирования предназначены для написания программ, которые потом выполняются ЭВМ автоматически. В таком слу-

79



чае говорят, что ЭВМ работает в программном режиме. Однако интерпретатор Бейсика дает возможность пользователю работать с ЭВМ и в так называемом командном режиме, когда ЭВМ выполняет каждую команду сразу же после ее ввода. В этом (и только в этом) режиме работают непрограммируемые микрокалькуляторы.

Перед тем, как дать команду, пользователь должен поинтересоваться, готова ли ЭВМ выполнять ее. О готовности к работе (другими словами — об ожидании команд от пользователя) интерпретатор Бейсика сигнализирует выдачей на экран соответствующего сообщения. Это может быть слово Ok (о'кей) или просто курсор в начале первой свободной строки. Курсор в различных ЭВМ указывает на то место на экране, где появится очередной введенный символ.

Пользователь печатает текст команды, нажимая соответствующие клавиши (как на пишущей машинке); текст появляется на экране. Когда вся команда набрана, надо нажатием специальной клавиши ввода сообщить об этом ЭВМ. До нажатия этой клавиши ЭВМ «не интересуется» тем, что пользователь написал на экране, а после нажатия приступает к анализу и выполнению команды. Важно помнить, что ввести информацию в ЭВМ — это значит не только набрать соответствующий текст, но и нажать после этого клавишу ввода.

Рассмотрим пример диалога пользователя и ЭВМ в командном режиме (текст, введенный ЭВМ, набран жирным шрифтом).

```
X=2
Y=-2
PRINT X+Y, X*Y
0 —4
X=-X ^ Y
PRINT X; 2*X; 3*X; 4*X
—,25 —,5 —,75 —1
PRINT 1/SQR (—X)
2
```

Если пользователь хочет, чтобы ЭВМ выполнила его программу, он должен сначала ввести эту программу в оперативную память. Рассмотрим простейший случай, когда пользователь в командном режиме вводит программу в ЭВМ строку за строкой. В Бейсике строки программ нумеруются. Наличие номера в начале строки, введенной пользователем — признак того, что это строка программы, которую надо только запомнить, добавив ее к тексту программы, который уже хранится в памяти. Так, если ввести в ЭВМ текст

```
PRINT «УРА!»
```

то на следующей строке экрана сразу появится

УРА!

Если же ввести текст

```
10 PRINT «УРА!»
```

то никакой видимой реакции не последует, однако в оперативной памяти после этого уже будет храниться программа, состоящая из одной этой строки. Если мы введем еще

```
20 PRINT «ПОНЯЛ!»
```

то в нашей программе будет уже две строки. Теперь прикажем ЭВМ выполнить эту программу. Для этого надо дать команду выполнения RUN (RUN — бежать). По ней ЭВМ приступит к выполнению программы, находящейся в оперативной памяти. В нашем случае в памяти хранится программа

```
10 PRINT «УРА!»
20 PRINT «ПОНЯЛ!»
```

и команда RUN вызовет печать строк УРА! ПОНЯЛ!

Затем будет выдано сообщение о готовности к выполнению следующей команды, и ЭВМ перейдет из программного режима в командный.

Таким образом, по команде RUN ЭВМ переходит из командного режима в программный, выполняет программу, переходит обратно в командный режим и ожидает следующую команду пользователя.

Вопросы.

1. В каких режимах может работать ЭВМ с интерпретатором Бейсика? (В командном и программном режимах.)

2. Как пользователь может дать команду ЭВМ? (Набрать текст команды, потом нажать клавишу ввода.)

3. Где на экране ЭВМ высветит следующий введенный пользователем символ? (В позиции, занимаемой курсором.)

4. Как можно заставить ЭВМ выполнить программу, находящуюся в оперативной памяти? (Надо дать команду RUN.)

Упражнения.

1. Что произойдет, если пользователь введет в ЭВМ строки

```
10 X=7
20 PRINT X; X*X; X^3
RUN
```

Ответ. ЭВМ сначала запомнит программу из двух строк, потом по команде RUN выполнит ее; в результате на экран будет выведена строка

```
7 49 343
```

Потом ЭВМ будет ожидать следующую команду.

2. Что произойдет, если пользователь введет в ЭВМ строки

```
10 A=5
20 B$=«BBB»
PRINT A; B$
```

Ответ. ЭВМ запомнит программу (первые

две строки), но не выполнит ее, а отпечатает 0 — переменным A и B значения еще не присвоены, поэтому A=0, а B\$=«».

## 9. Команда ввода информации

Для полноценного диалога с пользователем ЭВМ необходима возможность запросить нужную ей информацию. В простейшем случае это делается по команде

```
INPUT <имя переменной>
```

(INPUT — ввести). Выполняя ее, ЭВМ выводит на экран вопросительный знак и ожидает, пока будет введено значение указанной переменной. После нажатия пользователем клавиши ввода ЭВМ анализирует полученную информацию и присваивает соответствующее значение; команда считается выполненной, и ЭВМ переходит к выполнению следующей. Так, выполняя команду

```
100 INPUT X
```

ЭВМ будет ждать ввода значения переменной X. Если пользователь введет число, оно будет присвоено переменной X. Если же пользователь введет текст, который нельзя считать записью числа (например, ABC), то ЭВМ забракует эту информацию и потребует повторить ввод.

Соответственно, если в ответ на

```
120 INPUT A$
```

пользователь введет 123, то переменной A\$ будет присвоена символьная строка «123» (это не число 123!). При вводе символьной строки не обязательно заключать ее в кавычки.

Программист должен позаботиться, чтобы пользователь знал, чего от него ожидает ЭВМ, показывая ему вопросительный знак. Для этого можно, например, непосредственно перед командой ввода записать команду вывода с текстом, дающим пользователю соответствующие указания. Во многих версиях Бейсика, включая MSX, этот текст можно записать также непосредственно в команде INPUT. Например,

```
130 INPUT «Введите N»; N
```

выведет на экран строки

```
Введите N
```

```
?
```

и ЭВМ будет ждать ввода числа.

В одной команде INPUT можно запросить ввод значений для нескольких переменных. При этом имена переменных в команде следует разделить друг от друга запятыми; пользователь при вводе тоже должен разделять соответствующие значения друг от друга запятыми.

Рассмотрим пример.

```
NEW
10 PRINT «Введите X,A$»
20 INPUT X,A$
30 PRINT A$; X ^ 3
RUN
Введите X,A$
? 5 Куб
Куб 125
```

Ok

Отметим, что команду INPUT можно употреблять только в программе (в командном режиме проще дать значение переменной командой присваивания).

Вопросы.

1. Какие команды позволяют организовать диалог ЭВМ и пользователя? (Команды PRINT и INPUT.)

2. Как ЭВМ, выполняя команду INPUT, проверяет вводимую пользователем информацию? (Проверяется только соответствие типа введенного значения типу переменной.)

Упражнения.

1. Составить программу, которая потребует от пользователя значение для показателя степени X и выведет с объяснениями значение 2x.

Ответ. Возможная программа

```
10 PRINT «Введите показатель степени X»
20 INPUT X
30 PRINT «2 ^ » X; «=»; 2 ^ X
```

2. Составить программу, которая потребует ввести длины катетов и выведет длину гипотенузы.

Ответ. Возможна программа

```
10 PRINT «Введите катеты A, B»
20 INPUT X
30 PRINT «Гипотенуза»; SQR(A*A+
+B*B)
```

Окончание следует



## Занятия в КВТ: влияние на организм

Результаты различных исследований показывают, что работа за дисплеями вызывает напряжение центральной нервной и сердечно-сосудистой систем (ЦНС и ССС): зрительного анализатора и сопровождается влиянием различных физических факторов, источником которых может быть компьютер.

Поэтому нами изучалось влияние на функциональное состояние организма школьников IX классов комплекса условий при занятиях на компьютерах «Агат».

В процессе работы изучались: санитарно-гигиеническое состояние кабинетов ЭВТ (расположение и компоновка мебели, воздушно-тепловой режим, уровень освещенности и т. д.), функциональное состояние ЦНС и ССС, состояние зрительного анализатора, динамика умственной работоспособности, проводился хронометраж занятий и т. д.

Санитарно-гигиенические условия кабинетов ЭВМ обследованы в шести школах Москвы. Выявлена нерациональная расстановка мебели, отсутствие светорассеивающих устройств и т. п. Гигиенические показатели воздушно-теплого режима соответствуют в большинстве кабинетов нормативным данным, однако в течение дня часто отмечается его нарушение, что показывает необходимость тщательного контроля за санитарно-гигиеническим состоянием кабинетов.

Изменение уровня рентгеновского и электромагнитного излучений в разных направлениях на разных расстояниях от экрана показало, что в 50 см от экрана (нормированное минимальное расстояние) этот показатель значительно ниже предельно допустимой нормы и находится на уровне радиационного фона Земли. Проводится тщательный индивидуальный дозиметрический конт-

роль для выявления суммарных доз рентгеновского излучения (за 2 месяца, за год).

Изучение функционального состояния ССС, в частности сердечного ритма, показало, что у девятиклассников при работе на компьютерах выявляются изменения, свидетельствующие о значительном напряжении регуляторных механизмов. При этом выделены два типа регуляции ССС в зависимости от психо-эмоционального состояния школьников. Функциональная возможность ЦНС (по данным о латентном периоде и ошибочных действиях) при двухчасовой работе на компьютерах снижается.

Изучение функции ближнего зрения показало, что длительные занятия за дисплеями вызывают у школьников ослабление цилиарной мышцы, снижение устойчивости аккомодации. Функциональная лабильность зрительного анализатора также снижается.

Отмеченные изменения состояния зрительного анализатора через 3 месяца работы на «Агате» приводят к росту числа школьников с наличием остаточного напряжения аккомодации (так называемого спазма аккомодации), особенно у детей с близорукостью.

В целом полученные предварительные данные указывают на то, что занятия учащихся в дисплейных классах сопровождаются перестройкой работы основных функциональных систем организма. Особенно изменяется деятельность зрительного анализатора.

Работа учащихся при введении компьютерного обучения требует пристального внимания в плане гигиенического нормирования этих занятий и разработки профилактических мероприятий, направленных на охрану здоровья подростков.

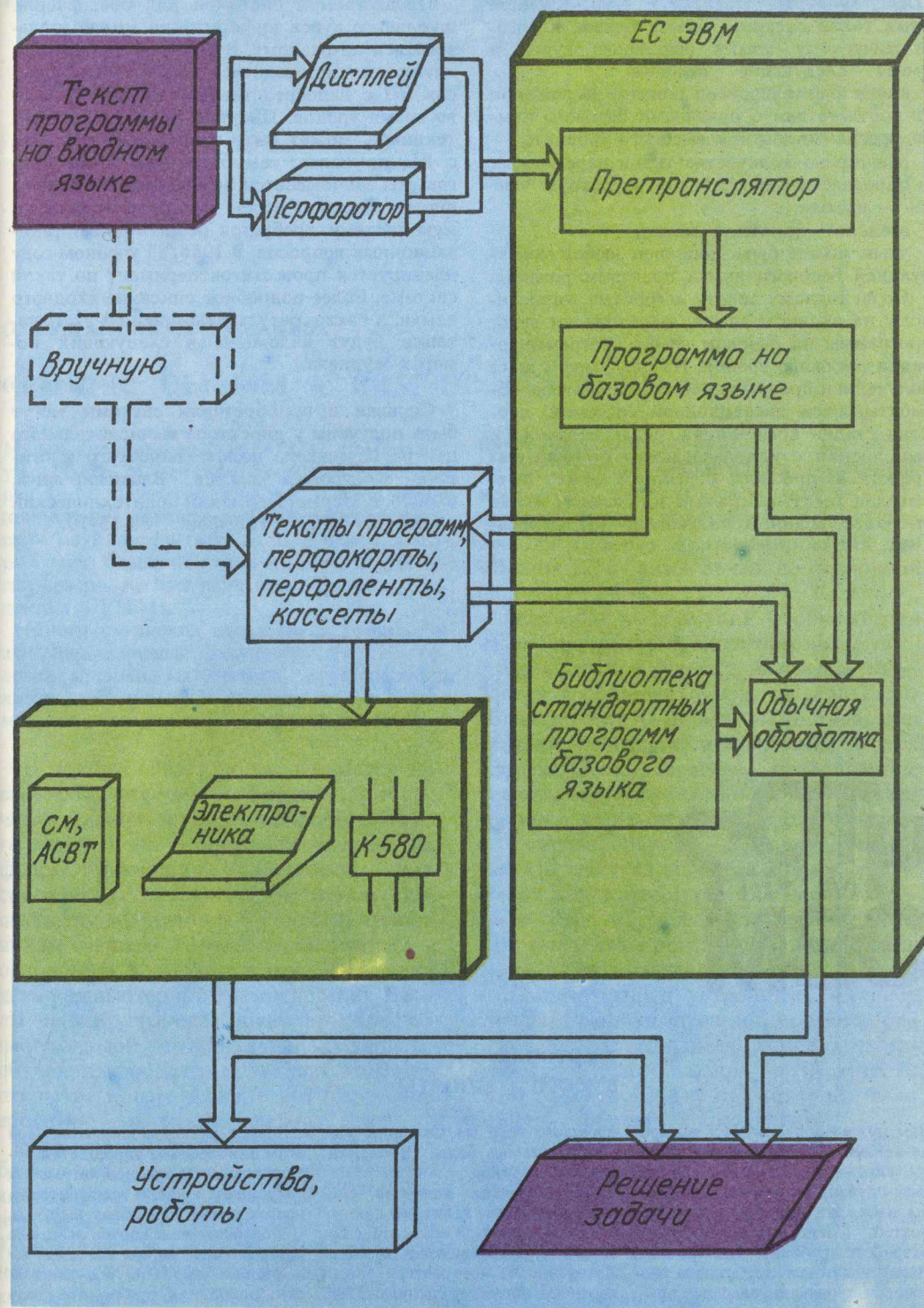
Л. ВИКЕНТЬЕВ, О. КОЗЛОВ, А. КОСТРОВ

## ЕС ЭВМ в курсе информатики

В ближайшее время планируется резко увеличить оснащение средних школ вычислительной техникой. Однако уже сегодня можно в школьной практике использовать парк вычислительных машин, в частности высокопроизводительных и распространен-

ных у нас в стране программно совместимых ЕС ЭВМ.

Авторы предлагают систему автоматического преобразования (претрансляции) алгоритмов, представленных на несколько расширенной версии школьного языка, в алго-





ритмы на одном из базовых языков программирования высокого уровня, таких, как ПЛ/1, Фортран, Паскаль и т. п. Входной язык такой системы весьма близок к алгоритмическому языку школьного курса и имеет следующие отличия:

в нем формализовано понятие выражения (в соответствии с правилами базового языка программирования высокого уровня);

расширено количество типов переменных; определены простейшие операторы ввода — вывода;

введено понятие метки; язык может быть дополнен любой конструкцией базового языка программирования.

По исходному тексту алгоритма, записанному на входном языке, формируется текст программы на базовом языке программирования высокого уровня. На следующем шаге полученная программа обычным образом обрабатывается операционной системой, проходя стадии компиляции, редактирования и выполнения с использованием стандартных средств. В процессе обработки может быть выведен текст программы на базовом языке высокого уровня. Использование претрансля-

тора позволит школьникам приобрести навыки будущего пользователя ЕС ЭВМ.

Предлагаемой системой для обеспечения школьного курса информатики можно пользоваться по-разному. В больших городах занятия могут проводиться в дисплейных классах ВЦ с непосредственной отладкой задач во время уроков. Школам, не обеспеченным техникой, может быть предложено общение с ВЦ по почте, тем более что текст программы записывается на обычном тетрадном листке. Однако, чтобы внедрить предлагаемую систему, требуется решить ряд организационных вопросов. В 1986/87 учебном году планируется провести эксперимент по такой системе. Более подробное описание входного языка, а также результаты пробной эксплуатации будут изложены в следующих номерах журнала.

Справки о приобретении системы могут быть получены у директора вычислительного центра Пермского политехнического института Кострова Алексея Владимировича, 614000, г. Пермь, Пермский политехнический институт, ВЦ. Телефон: 39-15-70.

## Внеклассная работа

О. ТУМАНОВА, Л. ЛИХАЧЕВА

### «Иначе бы мы к вам не пришли...»

Малая академия наук (УМАН) существует уже более 20 лет при Ухтинском индустриальном институте. Она была открыта по инициативе доктора геолого-минералогических наук профессора А. Я. Кремса, который внес большой вклад в развитие геологии севера. А. Я. Кремс был первым президентом УМАН.

Много ухтинских школьников прошло за эти годы основные факультеты УМАН: геологии, физики, математики, журналистики, химии, медицинский. Многим малая академия открыла дорогу к будущей профессии.

В ноябре 1985 г. у нас открылся новый факультет — информатики и вычислительной техники, деканом которого стала Л. М. Лихачева, сотрудник кафедры прикладной математики Ухтинского индустриального института. Цель открытия факультета — помочь школьникам нашего города ознакомиться с основными возможностями ЭВМ, овладеть компьютерной грамотностью. Школы нашего города пока не обладают необходимой материальной базой для обучения программированию, поэтому открытие нового факультета было своевременно и очень важно.

Наш индустриальный институт имеет более двух десятков вычислительных комплексов (ВК) «Искра-1256». На основе их создано три дисплейных класса. Кроме того, у нас есть ЭВМ ЕС-1022, на которой пока работают только в пакетном режиме.

Вычислительный комплекс «Иск-

ра-1256» содержит процессор, устройство отображения информации — дисплей, клавиатуру, печатающее устройство, графопостроитель, накопитель на магнитной ленте. Язык программирования для «Искры-1256» символьный со скобочной формой записи.

Вводимая информация отображается на экране дисплея. Если при вводе программы сделана ошибка, на экране появляется сообщение о ней. «Искра-1256» обладает богатыми возможностями для отладки программ, исправления в них ошибок. Есть средства, позволяющие стирать ненужные символы, целые предложения из памяти ЭВМ и одновременно с экрана, средства, позволяющие двигать текст на экране для того, чтобы вставить нужную букву или цифру. Предусмотрено и то, что пользователь может выводить программу и результаты счета не только на экран дисплея, но и на печатающее устройство, на лист бумаги. Графики функций, чертежи, рисунки можно выполнить, используя графопостроитель. Таковы возможности ЭВМ «Искра-1256», на которой было решено обучать школьников.

На факультет информатики и вычислительной техники записалось 70 школьников г. Ухты. Это были ученики VIII, IX, X классов. Их разбили на три группы. Занятия в группах вели преподаватели кафедр прикладной математики В. Е. Кондрашкин, В. Н. Лихачев и авторы этой статьи. Сначала школьни-

ЧТО  
МОЖЕТ  
ЭВМ

### Карточка вместо монеты

Вам нужно позвонить из телефон-автомата, а подходящей монеты, как назло, не найти. Типичный случай, не правда ли? Чтобы избежать подобных неприятностей, французские инженеры разработали новый электронный телефон-автомат, где оплата разговоров производится с помощью компьютерных карточек. В конце

прошлого года во Франции уже использовалось более 10 тыс. таких телефонов.

На карточках для них специальным кодом записаны данные о банковском счете абонента, с которого и удерживаются расходы. Телефонный компьютер, читая карточку, посылает сигналы ЭВМ, установленной в банке, и та реги-

стрирует расходы клиентов. При этом значительно упрощается обслуживание телефонов-автоматов, сокращается число инкассаторов, необходимых для сбора выручки.

Компьютерные карточки можно использовать не только для телефонной связи, но и в банковских операциях, при оплате покупок.



кам были прочитаны лекции по основам алгоритмизации и программирования, где рассматривались следующие вопросы: системы счисления, основные устройства ЭВМ, типы алгоритмов, блок-схемная запись алгоритмов, примеры линейных, разветвляющихся и циклических алгоритмов. Затем в дисплейных классах школьников познакомили с ЭВМ.

Первое время ребята играли с ЭВМ в такие игры, как «ХОККЕЙ», «ТЕННИС», «ФУТБОЛ», «МИННОЕ ПОЛЕ», «МОРСКОЙ БОЙ», «РЕВЕРСИ». Программы для этих игр были составлены заранее, записаны на магнитную ленту, и уже с ленты вводились в память ЭВМ. Любознательные и наблюдательные ребята, вызывая на экран игровую программу, рассматривали, как она построена, многие даже кое-что выписывали, чтобы использовать потом в своих программах.

После двух занятий с игровыми программами школьники уже знали, как включить ЭВМ, как подготовить ее к работе, ознакомились с клавиатурой. Таким образом, правильно организованная и содержательная игра много дает для развития ума и приобретения знаний. Хочется вспомнить слова русского марксиста Г. В. Плеханова: «Игра есть дитя труда, который необходимо предшествует ему во времени». Плеханов считал, что игра возникает в ответ на потребность общества в подготовке подрастающих поколений к жизни в обществе.

На втором этапе обучения мы ознакомили школьников с основными конструкциями языка программирования для «Искры-1256». Были приведены примеры составления программ, даны задания для самостоятельной работы.

Большое внимание на занятиях факультета информатики и вычислительной техники было уделено профессиональной ориентации школьников. При этом были использованы программы-тесты по профориентации, составленные студентами I курса нашего института С. А. Мельником и С. В. Лысенко.

Простейшая программа-тест помогает школьнику выбрать одну из пяти основных типов профессий: человек — природа, человек — техника, человек —

знак, человек — художественный образ, человек — человек. ЭВМ, работая по этой программе, выдавала на экран вопросы, связанные с каким-либо видом деятельности. Например: на экране появляется вопрос: «Любите ли Вы ухаживать за растениями?» Школьник, нажимая на клавиши, отвечал ей либо «ДА», либо «НЕТ». После того, как на все вопросы машина получит ответ, на экране появляется рекомендация. Например: «Вам подходит профессия «ЧЕЛОВЕК — ТЕХНИКА» или «ЧЕЛОВЕК — ПРИРОДА». Примечательно, что большинство школьников, посещающих наш факультет и испытываемых на тест «профессия», получили от машины ответ, что им больше всего подходит профессия «ЧЕЛОВЕК — ЗНАК», т. е. профессия, связанная с символами, знаками, с математикой, языками.

Когда мы, преподаватели, высказали по этому поводу свое удивление, школьники ответили, что ничего удивительного нет, «если бы было иначе, мы бы к вам не пришли». Занимаясь на факультете программирования со школьниками, мы, преподаватели, получали большое удовлетворение. Ребята удивляли нас своей заинтересованностью работой и программированием на ЭВМ, своей талантливостью в усвоении программирования. Многие школьники самостоятельно подбирали интересные задачи для составления программ, разработали и записали на магнитную ленту около десятка программ, реализующих различные тесты, игры, а также решения различных задач.

А. Хлыстов, ученик VIII класса школы № 2, составил две тестовые программы: «Определение типов характеров человека» и «Что вас беспокоит?». А. Забалуев, школа № 10, VIII класс, составил следующие программы: программа-шифратор букв; программа-игра «Лунолет-1», по которой ЭВМ выдает имитацию вертикальных маневров лунолета, программа-игра «Лунолет-2», в которой реализуется имитация горизонтальных и вертикальных маневров космического корабля, выхода на эллиптическую орбиту и стыковка со спутниками.

Г. Лукьянчиков, школа № 1, X класс, составил программы для проверки зна-

ний по химии, а также программу-игру «Орбитальная станция».

Ученики VIII класса школы № 3 Г. Касьянов, В. Тимотиевич, Д. Груздев разработали несколько игровых программ (таких, как «ХОД ФЕРЗЕЙ», «СЧАСТЛИВЫЕ БИЛЕТЫ» и др.). По своим разработкам они подготовили доклады на студенческую научно-техническую конференцию, которая проводилась в нашем институте в апреле 1986 г. Их доклады были заслушаны и одобрены на секции прикладной математики и вычислительной техники.

Подготовку таких выступлений по своим разработкам следует рассматривать как одну из форм приобщения школьников к началу научно-исследовательской работы.

Официально занятия для каждой группы проводились один раз в неделю по два часа. Однако большинство школьников приходило в дисплейные классы

## И. АНТИПОВ

НИИ школ Минпроса РСФСР

## Управляемый карандаш

Приобщить школьников к информатике можно во внеклассной работе, используя средства, сходные по действиям с простейшими графическими средствами языка Лого. Опыт работы Пионерского вычислительного центра (ПВЦ) — новой рубрики газеты «Пионерская правда» — показал, что даже учащиеся младших классов увлеченно составляли программы, по которым воображаемый персональный компьютер «Электроник» с помощью подключенного к нему «управляемого карандаша» выполнял построе-

еще 2—3 раза в неделю дополнительно, занимаясь самостоятельно, отлаживая свои программы и усовершенствуя их.

После окончания занятий, в мае слушатели факультета информатики и вычислительной техники — учащиеся X классов получили рекомендации-характеристики для поступления в институты. Остальные слушатели — учащиеся VIII и IX классов продолжают обучение программированию в следующем году. Мы предполагаем рассмотреть с ними язык программирования Фортран и поработать на ЭВМ ЕС-1022. Тех, кто придет к нам в первый раз, мы так же как и в этом году, обучим основам программирования и работе на «Искре-1256» в диалоговом режиме. Мы считаем, что занятия на нашем факультете УМАН дают многое для целеустремленных школьников, помогают преодолевать трудности.

ние разнообразных рисунков. Не только для построения рисунка, но и для решения какой-либо другой задачи на ЭВМ необходимо указать, какие команды и в каком порядке надо выполнять, т. е. составить программу для ЭВМ.

В соответствии с программой по заданию компьютера карандаш будет чертить на бумаге. Вот пример его работы. Карандаш установлен в центре прямоугольной системы координат. Рассмотрим несколько простейших команд для управления работой карандаша:

ТОЧКА (X, Y) —

команда установки карандаша в заданную точку.

По этой команде «Электроник» переставит карандаш с прежнего места в точку с указанными координатами X и Y;

— команды, предписывающие карандашу сдвинуться с прежнего места в указанном направлении на заданное число единиц (М клеточек на координатной сетке). При движении карандаш оставляет на бумаге след — линию.

ВВЕРХ М

ВНИЗ М

ВПРАВО М

ВЛЕВО М



При записи программы действий для удобства рекомендуется указывать порядковый номер команды в программе.

1. ТОЧКА (4,6) — установка карандаша в точку  $X=4$ ,  $Y=6$
2. ВЛЕВО 2 — построение «печатной» буквы С. После ее построения карандаш останавливается в точке  $X=4$  и  $Y=2$
3. ВНИЗ 4
4. ВПРАВО 2
5. ТОЧКА (6,2) — установка карандаша в точку  $X=6$  и  $Y=2$
6. ВПРАВО 2 — построение цифры 5
7. ВВЕРХ 2
8. ВЛЕВО 2
9. ВВЕРХ 2
10. ВПРАВО 2

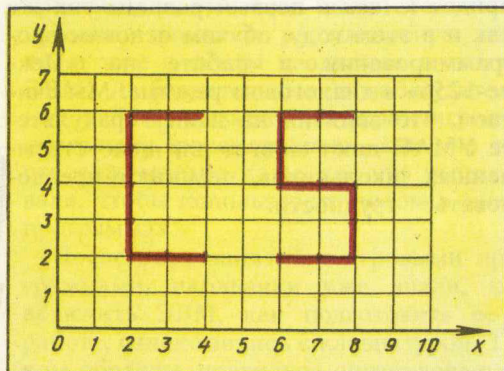


Рис. 1

Упражнения. 1. Как изменится программа, если сначала нарисовать цифру 5, а затем букву С, т. е. «5С»?

2. Указать координаты точки, в которой остановился карандаш после выполнения команды № 10.

3. Отметить на рисунке точки, задаваемые следующими командами:

1. ТОЧКА (1,1)
2. ТОЧКА (3,5)
3. ТОЧКА (5,1)

4. По какой команде карандаш вернется в начало системы  $XOY$ ?

5. Что будет нарисовано в результате исполнения программы:

1. ТОЧКА (3,3)
2. ВВЕРХ 5
3. ВПРАВО 5
4. ВНИЗ 5
5. ВЛЕВО 5

6. Составить программу, по которой будет нарисован квадрат со стороной в 5 единиц и нижней левой вершиной в начале координат.

7. Составить программу, по которой будет нарисован прямоугольник.

Для примера составим программу, по которой будет нарисована буква С и цифра 5 (рис. 1).

8. Составить программу для изображения, указанного на рис. 1. При этом букву С начинайте рисовать с точки  $X=4$  и  $Y=2$ , а цифру 5 с точки  $X=8$  и  $Y=6$ .

9. Как изменится программа, если те же С5 нарисовать смещенными на 3 единицы вправо и 3 единицы вверх?

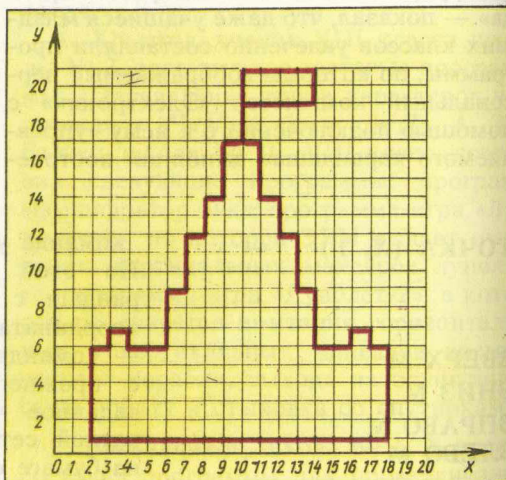
10. Составить программу для изображения на рис. 2.

Условимся, что перед выполнением новой программы карандаш автоматически устанавливается в начало координат.

После усвоения учащимися простых команд можно перейти к командам проведения отрезка и окружности, позволяющих строить более сложные рисунки. Команды имеют вид:

ОТРЕЗОК ( $X$ ,  $Y$ ) — (сокращенно ОTR  $X,Y$ )

Рис. 2



ОКРУЖНОСТЬ  $P$  — (сокращенно ОКР  $P$ )

Команда проведения отрезка означает соединение прямой линией точки, в которой находится карандаш, и точки с указанными координатами  $X,Y$ . Команда проведения окружности означает рисование окружности с центром в точке нахождения карандаша до выполнения команды и радиусом  $P$  единиц. После этого карандаш остается в центре окружности.

Нетрудно заметить, что с помощью команды проведения отрезка можно выполнять те же действия (построения), что и с командами продвижения карандаша вверх—вниз—вправо—влево на заданное число единиц. Командами ВВЕРХ  $M$ , ВНИЗ  $M$ , ВПРАВО  $M$ , ВЛЕВО  $M$  удобно пользоваться, но без них можно обойтись, пользуясь командой ОТРЕЗОК ( $X$ ,  $Y$ ).

Упражнения. 1. В одной из ранее написанных программ, например для построения изображения на рис. 1, замените команды вверх—вниз—вправо—влево на команды проведения отрезка. Какой вариант предпочтительнее?

2. Что будет нарисовано при исполнении следующей программы:

1. ТОЧКА (2,1)
2. ОКР 1
3. ОTR (6,1)
4. ОКР 1
5. ОTR (4,4)
6. ОКР 1
7. ОTR (2,1)

3. По данному рисунку составить программу. Составить программу, по которой можно нарисовать велосипед (модели могут быть разными).

4. Сколько команд потребуется для написания слова МИР?

5. Придумайте одну или несколько новых команд для карандаша.

Рассмотрим составление программы и построение изображения скворечника (рис. 3). В его изображении выделяют составные части: стены и пол, окошко, ступенька, крыша. Сначала необходимо выбрать приблизительное расположение скворечника относительно начала координат и его размеры. Для этого надо сделать эскиз. Начать построение можно с любой части рисунка. При этом будут

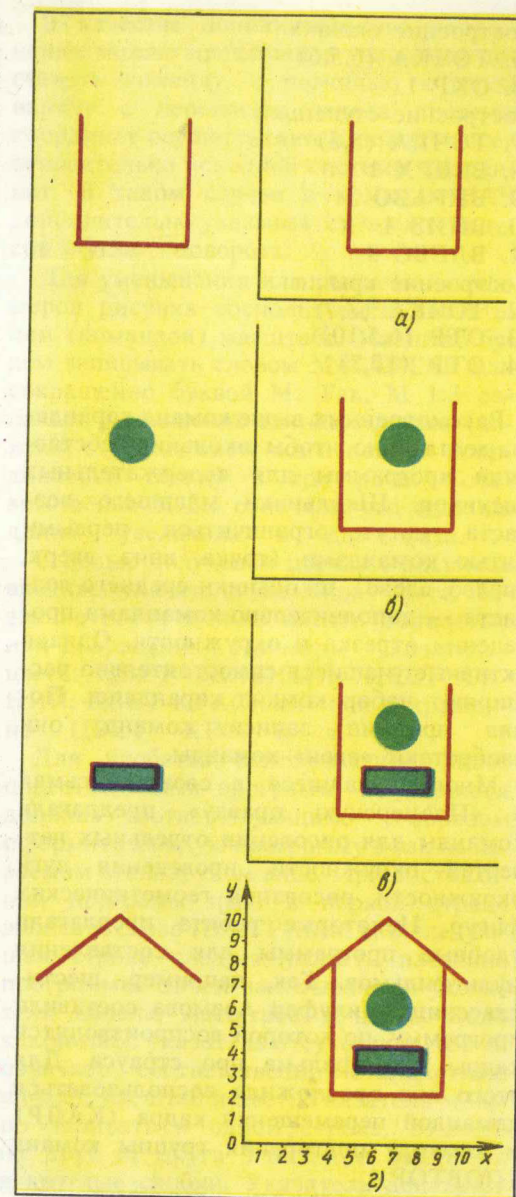


Рис. 3

возникать соответствующие варианты программы. Для иллюстрации рассмотрим один из вариантов программы. На рис. 3 отмечены этапы построения данного изображения.

Программа:

Построение стен и пола

1. ТОЧКА (4,8)
2. ВНИЗ 6
3. ВПРАВО 5
4. ВВЕРХ 6



Построение окошка

5. ТОЧКА (6,5,6)

6. ОКР 1

Построение ступеньки

7. ТОЧКА (5,3)

8. ВВЕРХ 1

9. ВПРАВО 3

10. ВНИЗ 1

11. ВЛЕВО 3

Построение крыши

12. ТОЧКА (3,7)

13. ОТР (6,5,10)

14. ОТР (10,7)

Рассмотренных выше команд карандаша достаточно, чтобы школьники составляли программы для содержательных рисунков. Школьники младшего возраста могут ограничиться первыми пятью командами (точка, вниз, вверх, вправо, влево), школьники среднего возраста — дополнительно командами проведения отрезка и окружности. Однако активные учащиеся самостоятельно расширяют набор команд карандаша. Поняв принцип записи команд, они изобретают «свои» команды.

Многие учащиеся в своих письмах в «Пионерскую правду» предлагали команды для рисования отдельных четвертей окружности, проведения дуги окружности, рисования геометрических фигур. Некоторые ребята предлагали удобные программы для составления мультфильмов. Так, например, шестиклассница Нилуфар Азамова составила программу, по которой воспроизводятся кадры мультфильма про страуса. Для этого она предложила воспользоваться командой перемещения кадра (КАДР) и команду повторения группы команд (ПОВТОР).

За словами КАДР и ПОВТОР в соответствующих командах записываются числа, указывающие смещение кадра, номера повторяемых команд программы. Не следует стремиться увеличивать число новых команд, но интересные предложения необходимо обсудить, проверить правильность предлагаемой команды и удобства ее использования. Здесь большой простор для творчества школьников.

Для рисования одной и той же фигуры можно предложить разные программы. Может быть различный порядок вы-

полнения тех или иных деталей контура фигуры, использование разных команд для проведения линий. Однако следует отметить, что лучшие программы позволяют выполнить построение с наименьшим числом команд. Такие программы можно назвать оптимальными. Чтобы их составить, надо проанализировать последовательность действий для получения изображения рисунка. Наименьшее число команд получается, когда стремятся сэкономить количество команд по перестановке карандаша из точки в точку (прерывание линий), стараются изобразить контур без отрыва карандаша, избежать лишнего обвода линий. Так, для изображения пятиконечной звезды получается оптимальная программа из 6 команд (установка в начальную точку и проведение без отрыва пяти отрезков).

Упражнения. 1. Убедитесь, что изображение одной и той же фигуры получается при исполнении каждого из следующих вариантов программы:

- |   |                |                |
|---|----------------|----------------|
| 1. ТОЧКА (2,4)  | 1. ТОЧКА (2,4) | 1. ТОЧКА (2,1) |
| 2. ОКР 1  | 2. ОКР 1       | 2. ВВЕРХ 3     |
| 3. ТОЧКА (2,1)  | 3. ВНИЗ 3      | 3. ОКР 1       |
| 4. ОТР (2,4)  |                |                |
| 2. Рассмотрите два варианта программы для изображения буквы «А». Исполните эти программы. За счет чего достигнута экономия? |                |                |
| 1. ТОЧКА (2,1)  | 1. ТОЧКА (2,1) |                |
| 2. ОТР (3,4)  | 2. ОТР (4,7)   |                |
| 3. ВПРАВО 2   | 3. ОТР (6,1)   |                |
| 4. ОТР (6,1)  | 4. ТОЧКА (3,4) |                |
| 5. ТОЧКА (3,4)  | 5. ВПРАВО 2    |                |
| 6. ОТР (4,7)  |                |                |
| 7. ОТР (5,4)  |                |                |

3. Постройте фигуру, выполняя следующие действия:

а) Проведите окружность с центром в точке  $X=5$  и  $Y=16$ ; радиус окружности — 1 (одна клеточка).

б) Постройте прямоугольник, вершинами которого являются точки с координатами

$X=4, Y=5$ ;  
 $X=4, Y=15$ ;  
 $X=6, Y=15$ ;  
 $X=6, Y=5$ .

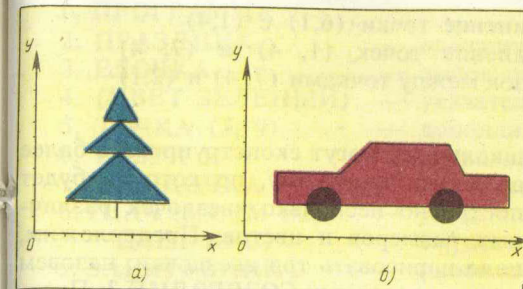


Рис. 4

в) Соедините отрезками точку  $X=5, Y=2$  с точками

$X=4, Y=5$ , и  $X=6, Y=5$ .

Составьте программу с наименьшим числом команд (9 команд).

4. Какое наименьшее число команд потребуется для программы построения елочки, изображенной на рис. 4а?

5. Для построения контура легкового автомобиля, изображенного на рис. 4б, оптимальная программа состоит из 15 команд. Составьте такую программу (рассмотрите разные способы построения).

С целью более углубленного изучения темы для построения рисунков можно ввести указатели расположения начала координат, масштаба, цвета.

Когда требуется одну и ту же фигуру нарисовать в какой-либо другой части координатной сетки, то приходится изменять координаты точек, указанные в подготовленной ранее программе. Эту работу можно упростить, если воспользоваться новой командой, указывающей, где будет расположено начало координатной сетки, в которой должно быть выполнено построение нужной фигуры.

НАЧАТЬ ( $X, Y$ ) или сокращенно НАЧ ( $X, Y$ )

Например, команда НАЧ (3, 8) означает, что построение фигуры будет осуществляться в системе координат, центре которой расположен в точке с координатами  $X=3$  и  $Y=8$ . Данную команду будем называть указателем начала.

- |                |                                    |
|----------------|------------------------------------|
| 1. ПРОГРАММА   | — начало программы                 |
| 2. ЗВЕЗДОЧКА   | — название программы               |
| 3. ТОЧКА (2,1) | — команды построения звездочки:    |
| 4. ОТР (4,6)   | — соединение точки (2, 1) с (4, 6) |
| 5. ОТР (6,1)   | — соединение точек (4,6) и (6,1)   |

В качестве дополнительного упражнения можно предложить учащимся составить команду, с помощью которой наряду с переносом начала системы координат осуществляют и поворот осей относительно исходной системы координат. В таком случае в команде НАЧ дополнительно указывается число градусов угла поворота.

Для уменьшения или увеличения размеров рисунка воспользуемся указателем (командой) масштаба. Масштаб будем записывать словом МАСШТАБ или сокращенно буквой М. Так, М 1:3 означает уменьшение единицы длины (каждой стороны клеточки исходной координатной сетки) в три раза. Соответственно М 2:1 означает, что единица длины увеличена в два раза.

С целью получения цветных рисунков вводится указатель (команда) цвета. Название цвета можно писать полностью или сокращенно. Примеры указателей цвета: ЦВЕТ ЗЕЛЕНый; ЦВЕТ СИНИй; ЦВЕТ КРАСНый или сокращенно ЦВЕТ КРС и т. п.

Для удобства использования ранее подготовленных программ учащиеся должны соответствующим образом оформить каждую программу в так называемом стандартном виде. При составлении программ надо в первой строке записать слово ПРОГРАММА. В следующей строке следует указать название программы. Затем записываются указатели начала, масштаба и цвета. Если какой-либо указатель отсутствует, то это означает соответственно стандартные начало, масштаб и цвет. Указатели можно записать в одной строке, отделив их друг от друга запятой и заключив в круглые скобки. Указатель действует до конца программы. В последней строке программы записывается слово КОНЕЦ.

Рассмотрим пример оформления программы построения пятиконечной звезды, назовем эту программу ЗВЕЗДОЧКА



6. ОТР (1,4) — соединение точки (6,1) с (1,4)
7. ОТР (7,4) — соединение точек (1, 4) и (7, 4)
8. ОТР (2,1) — отрезок между точками (7, 4) и (2,1)
9. КОНЕЦ

Учащиеся проявляют творчество и предлагают использовать свои способы записи последовательности соединения точек отрезками. Например, для приведенной выше программы:

X	2	4	6	1	7	2
У	1	6	1	4	4	1

(2,1) — (4,6) — (6,1) — (1,4) — (7,4) — (2,1) и т. п. Используя ранее составленную программу ЗВЕЗДОЧКА,

школьники могут сконструировать более сложную программу, по которой будет построено несколько звездочек различных размеров и цветов. Предположим, надо нарисовать три звездочки; назовем новую программу СОЗВЕЗДИЕ 3. Пусть одна из них будет стандартного размера (в соответствии с приведенной выше программой) и зеленого цвета, вторая — в два раза меньше и синего цвета, третья — в два раза больше и красного цвета (рис. 5).

1. ПРОГРАММА
2. СОЗВЕЗДИЕ 3 — название программы
3. ЗВЕЗДОЧКА — вызов (обращение) программы ЗВЕЗДОЧКА
4. (ЦВЕТ ЗЕЛЕНЫЙ) — указатель цвета — зеленый (см. рис. 5а)
5. ЗВЕЗДОЧКА — вызов программы ЗВЕЗДОЧКА (см. рис. 5б)
6. (НАЧ (14,1), М 1:2, ЦВЕТ СИНИЙ) — указатели: начало новой координатной сетки  $X=14$ ,  $Y=1$ ; уменьшение в 2 раза, цвет — синий
7. ЗВЕЗДОЧКА — вызов программы ЗВЕЗДОЧКА (см. рис. 5в)
8. (НАЧ (4,10), М 2:1, ЦВЕТ КРАСНЫЙ) — указатели: начало  $X=4$ ,  $Y=10$ , увеличение в два раза, цвет красный
9. КОНЕЦ

Здесь ранее составленная программа построения звездочки выступает в роли подпрограммы, к которой в новой про-

грамме СОЗВЕЗДИЕ 3 происходит обращение с разными параметрами — указателями начала, масштаба и цвета.

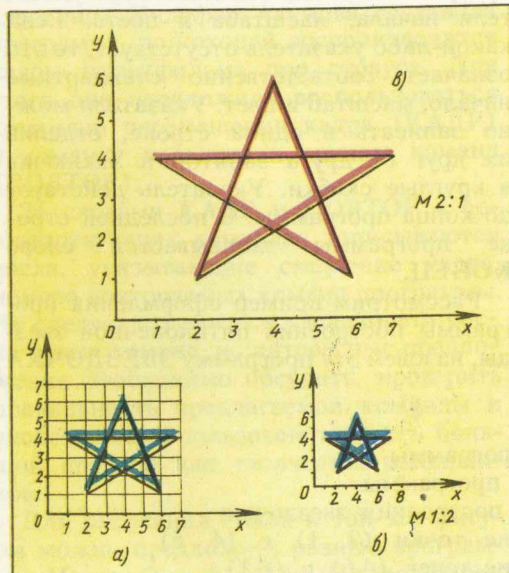
**Упражнения. 1.** Составить программу, по которой будет нарисован робот. Программу оформите в стандартном виде и назовите ее ЭЛЕКТРОНИК.

**2.** Составьте сложный рисунок, на котором изображены ЕЛОЧКА, ЗВЕЗДОЧКА, ЭЛЕКТРОНИК.

**3.** Какими программами (подпрограммами) удобно воспользоваться для построения рисунка из предыдущего упражнения?

**4.** Выполните построения в соответствии с программой ПРАЗДНИК, приведенной ниже, в которой используются подпрограммы ЕЛОЧКА, ЗВЕЗДОЧКА, ЭЛЕКТРОНИК. В строке 10 в круглых скобках запишите подобранные вами указатели, в соответствии с которыми надо нарисовать ЭЛЕКТРОНИКА у елки.

Рис. 5



1. ПРОГРАММА
2. ПРАЗДНИК
3. ЕЛОЧКА
4. (ЦВЕТ ЗЕЛЕНЫЙ)
5. ТОЧКА (5, 9)
6. ВВЕРХ 1
7. ЗВЕЗДОЧКА
8. (НАЧ (3,9), М 1:2, ЦВЕТ КРС)
9. ЭЛЕКТРОНИК

- название конструируемой программы
- вызов подпрограммы ЕЛОЧКА
- указатель цвета — зеленый
- дополнительные построения (для установки звездочки на елке)
- вызов подпрограммы ЗВЕЗДОЧКА
- указатели: начало  $X=3$  и  $Y=9$ , уменьшение в 2 раза, цвет — красный
- вызов подпрограммы построения фигуры, с соответствующими указателями начала, размера и цвета

10. (.....)
11. КОНЕЦ

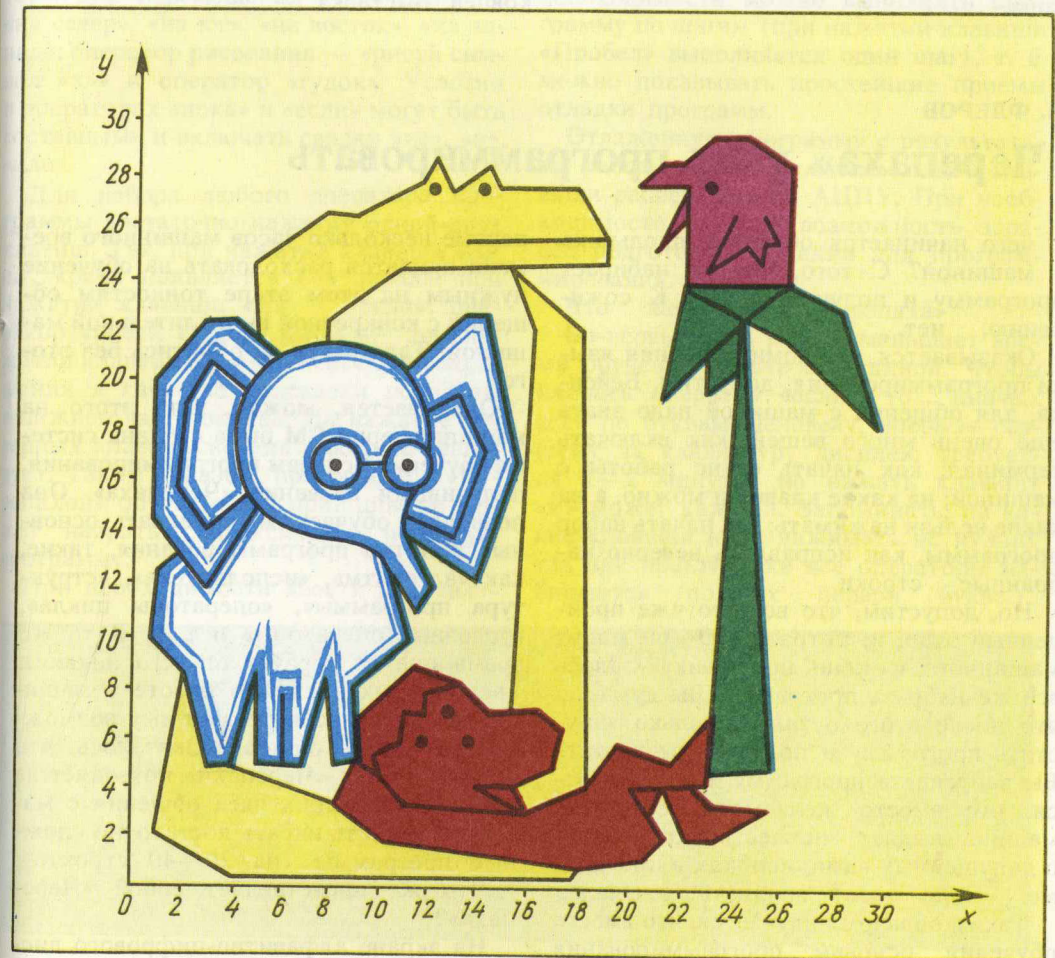
**5.** Сконструируйте свою картину из отдельных фигур по готовым или составленным вами программам, запишите общую программу.

На рис. 6 приведен рисунок, полу-

ченный по программе, составленной шестиклассницей Олей Кириленко из Подмоскovie. Сюжетом для ее рисунка послужил мультфильм «38 попугаев».

Участвуя в работе ПВЦ или матема-

Рис. 6





тического кружка по данной теме, учащиеся знакомятся с элементами информатики. Осваивая простейшие графические средства и приемы программирования, школьники готовятся к изучению курса информатики в IX—X классах.

Ниже приводится примерный тематический план, по которому руководители кружков могут проводить занятия по основам компьютерной графики, используя материалы ПВЦ газеты «Пионерская правда».

#### Тематический план занятий

1. ЭВМ и программирование. Исполнитель команд. Простейшие команды управляемого карандаша.
2. Команды проведения отрезков и окружностей.
3. Дополнительные команды карандаша.

М. ФЛЕРОВ

## «Черепашка» учит программировать

С чего начинается общение школьника с машиной? С того, что он набирает программу и получает ответ? К сожалению, нет.

Оказывается, что помимо знания языка программирования, допустим, Бейсика, для общения с машиной надо знать еще очень много вещей: как включать терминал; как начать сеанс работы с машиной; на какие клавиши можно, а на какие нельзя нажимать; как начать набор программы, как исправлять неверно набранные строки.

Но, допустим, что все это уже пройденный этап, и, потратив 30—60 минут машинного времени, школьнику удалось все же набрать программу. Вы думаете, что после этого осталось только запустить программу и получить результат? Вы запускаете программу, и оказывается, что вместо желаемого результата машина выдает множество сообщений о допущенных вами ошибках в программе. Надо все начинать сначала.

Таким образом получается, что вместо обучения основам программирования

4. Перенос координатной сетки (смещение).

5. Оформление программы построения рисунка.

6. Указатель масштаба. Рисунки в заданном масштабе.

7. Указатель цвета. Цветные рисунки.

8. Указатель толщины линий.

9. Повторение группы команд. Команда КАДР. Простейшие мультфильмы.

10. Организация библиотеки программ.

11. Конструирование сложных программ.

12. Коллективное выполнение программы. Построение сложного рисунка.

13. Индивидуальные занятия. Подготовка работ к выставке.

14. Машинная распечатка рисунков. Дисплей. Графопостроитель.

15. Экскурсия на ВЦ.

16. Обсуждение лучших работ кружковцев. Выставка.

первые несколько часов машинного времени придется расходовать на обучение нужным на этом этапе тонкостям общения с конкретной вычислительной машиной. Так нельзя ли обойтись без этого?

Оказывается, можно. Для этого на машинах серии СМ была создана система обучения основам программирования, получившая название «Черепашка». Она позволяет обучаемому осваивать основные понятия программирования, такие, как «алгоритм», «исполнитель», «структура программы», «операторы цикла», «условные операторы» и т. д. В то же время она не требует от него никаких специальных знаний о работе с машиной и практически не дает ему возможностей делать ошибок. Оказалось, что использование «Черепашки» позволяет за два академических часа общения с машиной научить писать достаточно сложные программы (на 20—40 строк).

Что же представляет собой «Черепашка»?

На экране алфавитно-цифрового дис-

плея обучаемый видит поле, огражденное с краев звездочками. В углу поля находится маркер — «Черепашка». Система позволяет писать программу управления «Черепашкой» на языке, максимально приближенном к языку программирования, описанному в школьном учебнике информатики. Под управлением написанной программы «Черепашка» может перемещаться по полю вверх, вниз, направо, налево, рисовать любые символы на поле, проверить, какой символ находится под ней, и менять свои действия в зависимости от этого.

Программа для управления «Черепашкой» пишется в левой стороне экрана. Язык программирования включает такие знакомые по учебнику информатики конструкции, как: «алг... нач... кон», «если... то... иначе... все», «пока... нц... кц». В язык включены специальные операторы для перемещения «Черепашки» — «на север», «на юг», «на восток», «на запад»; оператор рисования — «рисуй символ «х»» и оператор «гудок». Условия в операторах «пока» и «если» могут быть составными и включать связки «не», «и», «или».

Для набора любого оператора программы достаточно нажатия одной-двух клавиш. Так, при нажатии клавиши «З», на экране появляется «на запад»; при нажатии клавиши «Е» — «если»; оператор «рисуй символ «+»» — при нажатии клавиш «Р» и «+». Все эти сокращения в качестве подсказки показаны в нижней части экрана. На нажатие неверных клавиш система никак не реагирует. Таким образом, при работе с «Черепашкой» обучаемый в принципе не может набрать синтаксически неверную программу.

При необходимости внести исправле-

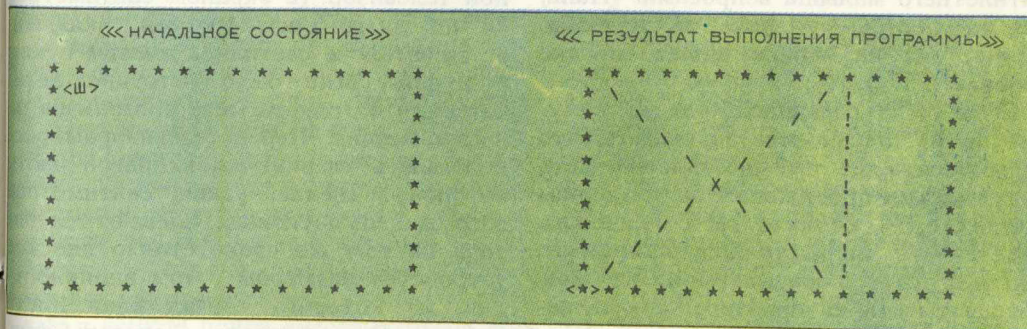
ния в программу достаточно подвести курсор к ненужной строке программы, нажать клавишу «удаление», после чего набрать нужный оператор (конечно же, 1—2 нажатиями на клавиши). «Черепашка» сразу приучает к структурному программированию. Так, структурные операторы, например структура «если... то... иначе... все», создаются целиком при нажатии клавиши «Е» и удаляются тоже только целиком. Структуры в программе появляются на экране с необходимыми для красивого оформления отступами. Набрать неструктурную или некрасиво оформленную программу невозможно.

После набора программу можно запустить на выполнение. При этом видно, как «Черепашка» по шагам выполняет программу. Темп выполнения выбран так, что за ним можно уследить. При необходимости можно выполнять программу по шагам (при нажатии клавиши «Пробел» выполняется один шаг), т. е. можно показывать простейшие приемы отладки программ.

Отлаженную программу с результатами работы можно нажатием одной клавиши распечатать на АЦПУ. При необходимости имеется возможность заранее подготовить задания для программирования.

Что же дает «Черепашка»?

Во-первых, она резко уменьшает время обучения работе с машиной. Чтобы набрать оператор «если... то... иначе... все» по буквам, человеку, впервые севшему за клавиатуру дисплея, требуется 1—2 минуты, но нажать клавишу «Е» может каждый. Запоминать, на какие клавиши надо нажимать, не нужно, так как практически все операторы набираются по их первой букве.





Во-вторых, работать с «Черепашой» начинающему значительно интереснее, чем, например, писать программу на Бейсике, производящую какие-либо численные расчеты. Все и так знают, что машина умеет работать с числами. А вот писать программы, рисующие на экране картинки, куда интереснее.

Надо конечно учитывать, что «Черепаша» предназначена только для изучения основных понятий программирования. Использовать ее стоит только для первых 4—6 часов занятий на машине.

Привожу пример простейшей программы для «Черепашки» (в том виде, в каком она выдается на печать).

#### Программа

```
нач
. пока не « * » нц
. . рисуй символ « \ »
. . на юг
. . на восток
. кц
```

```
. на север
. пока « » нц
. . рисуй символ «!»
. . на север
. кц
. на юг
. на запад
. делай «И»
. гудок
кон
алг «И»
нач
. пока не « * » нц
. . если « » то
. . . рисуй символ «/»
. . иначе
. . . рисуй символ «х»
. . все
. . на запад
. . на юг
. кц
кон
```

С. ПАЧИКОВ

## Дома за дисплеем

Я хочу рассказать о том, что компьютер — это совсем не страшно, что он может походить не на шкаф, утыканный мигающими лампами, а на пишущую машинку, что персональным домашним компьютером и готовыми программами может пользоваться любой человек.

Мы с моим знакомым сидим у него на кухне, обсуждая проблемы компьютерной графики. Поминутно забегают его сын Саша с довольно странными для пятилетнего малыша вопросами: «Папа, как мне программу с дискеты стереть?», «Папа, она на экране пишет синтаксис эррор..» и т. п.

Если бы в середине 70-х гг., кому-нибудь из программистов сказать, что через десять лет пяти-шестилетние дети будут часами просиживать за домашним компьютером, сочиняя свои маленькие, простенькие, но настоящие программы, что они будут «сбрасывать» их на диск, распечатывать на принтере, что за несколько секунд ребенок сменит про-

грамму, и из шахматного партнера компьютер превратится в какой-нибудь, по его выбору, музыкальный инструмент или в тренажер, обучающий вождению самолета, автомобиля, что стоимость домашнего компьютера будет сравнима с ценой цветного телевизора...

А вот Саша прибежал с очередным вопросом, и я прошу его показать мне, чем он занимается.

Саша провел меня в кабинет своего отца. На столе стоял небольшой цветной телевизор (с экраном сантиметров 30 по диагонали). Он был включен, на синем фоне светились желтые буквы и цифры. Мне было нетрудно догадаться, что это фрагмент программы на языке Бейсик. Перед телевизором располагалась клавиатура, в правой части которой виднелась узкая, сантиметров восемь, щель дисковода. Слева от телевизора на том же столе стояло плоское устройство размером с том энциклопедии, из которого торчал край листа бумаги. «Это монитор», — сказал Саша,

показывая на телевизор. «Это компьютер с дисководом», — жест в сторону «клавиатуры». «А это принтер. Папа на нем статьи научные печатает». Я попросил мальчика показать мне все с самого начала. «Сейчас», — сказал Саша, — я только запишу свою программу». Он залез на кресло с ногами, быстро набрал на клавиатуре команду SAVE MAMA (что на большинстве языков программирования означает — записать ту программу, которая сейчас в памяти компьютера, на внешний носитель, в данном случае на маленький магнитный диск, присвоить программе имя MAMA и занести это имя в каталог, который хранится на том же диске). Набранная команда тут же появилась на экране. Саша нажал на клавиатуре большую клавишу с надписью ENTER (вести, выполнить). Около щели дисковода загорелась красная лампочка, послышалось тихое жужжание. Через несколько секунд лампочка погасла, звук прекратился. На экране, чуть ниже последней надписи, появилось слово READY (готово). (Я так подробно описываю действия Саши еще и потому, что эта процедура типична для любого ПК, как отечественного, так и зарубежного.) Саша вынул дискету из дисковода и вставил на ее место другую. «Показать вам, какие у меня есть игры?» — и, не дожидаясь ответа, набрал на клавиатуре слово CAT. «Кат — это каталог», — пояснил он. На экране появился перечень программ, содержащихся на дискете. Я увидел в списке шахматы и попросил показать их мне. Несколько ударов по клавишам — и на экране появилась шахматная доска с расставленными фигурами. Компьютер предложил выбрать желаемый цвет фигур. «Саша, ты умеешь играть в шахматы?» — «Да, умею — плохо. Все время проигрываю. Если букву Р нажать, то компьютер подсказывает, как пойти, но я сейчас редко спрашиваю». Саша выбрал белые фигуры и сделал ход Остапа Бендера — e2—e4. Компьютер стал разыгрывать «сицилианку». Типичная шахматная программа хранит в памяти несколько сот дебютов, и если вы играете «по теории», то на первые десять — двенадцать ходов компьютер отвечает мгновенно, но стоит вам в дебюте пойти «нестандартно» или

после того, как она исчерпает «дебютную заготовку», машина начинает задумываться. Время размышления компьютера над очередным ходом зависит от того, какой уровень игры вы ему заказали. Саша выбрал самый простой, «нулевой», уровень, поэтому компьютер задумывался не более чем на пять-шесть секунд. И хотя машина играла слабо, на пятнадцатом ходу Саша получил мат. Вообще, прогресс в шахматных программах впечатляет. Существуют программы для ПК, выполнившие норму мастера спорта.

Саша, а я знаком с ним давно, обычный смысленный пятилетний мальчик, такой же, как и тысячи его сверстников. И то, что он довольно легко оперирует с компьютером, заслуга не только и не столько его, сколько компьютера. Правда, то, что Саша вообще с техникой «на ты», — заслуга его родителей: я никогда не слышал, чтобы они говорили детям: «Не трогай — сломаешь!»

Современный бытовой компьютер (кстати, название советской микро-ЭВМ БК-0010 означает: бытовой компьютер, модель 0010) ориентирован на людей, не имевших опыта общения с ЭВМ. Никаким произвольным нажатием клавиш сломать его невозможно. Если вы делаете не то — он либо скажет, что он вас не понимает, либо сообщит, какого рода ошибку вы совершили. Модели более современные и более дорогие отличаются от более простых не только быстродействием и объемом памяти, но и способностью облегчать общение с собой. Это не означает, что домашний компьютер — нечто чрезвычайно простое. Но ПК устроены таким образом, чтобы их использование и более глубокое освоение шли параллельно. Для того чтобы научиться его включать и «загружать» готовую программу, достаточно один раз прочесть инструкцию. Чтобы научиться большему — работать с «редактором текста» (программа, позволяющая набирать любой текст, редактировать его, сохранять на диске, распечатывать на принтере), «базой данных» (система хранения и поиска разнообразной информации) и им подобными программами, потребуется несколько дней. По мере работы вы будете осваивать



их лучше и глубже. Научиться основным навыкам программирования можно за несколько недель. Совершенствоваться в программировании можно всю жизнь. Процесс этот необычайно увлекателен. В момент, когда даже несложная программа, которую ты сам написал, вдруг начинает работать и на экране появляется замысловатый геометрический рисунок, или танцующий мышонок, или график нехитрой функции, или при каждом нажатии на клавишу встроенный в компьютер динамик начинает издавать звук, который ты сам «сконструировал», испытываешь восторг и удивление.

«Саша, ты читать умеешь?» — «Да, умею — плохо». — «А писать?» — «Умею — письмо дедушке написал. Показать?» — он «сбросил» шахматы и вызвал на экран каталог диска. Отыскав в каталоге нужную программу с довольно странным названием ПИСМО, он загрузил ее в память и запустил. На экране появился текст «Здравствуй дорогой дедушка. Я очен по тебе скучаю. Приезжай скорее». — «Саша, ты мягкий знак пропустил в слове «очень». Саша отыскал глазами слово и, используя четыре клавиши с нарисованными на них стрелками (вверх, вниз, вправо, влево), стал передвигать по экрану маленький, размером с букву, мерцающий квадратик, именуемый курсором. Когда курсор оказался справа от слова «очен», Саша вставил недостающую букву, и весь последующий текст аккуратно сдвинулся вправо. Истины ради скажу, что на столе лежал лист бумаги, на котором тот же текст был написан фломастером, большими печатными буквами, без этой ошибки. «Что у тебя еще есть?» — «Ну, игры разные, например, чтобы учиться время по часам узнавать, там такой человечек, он, когда правильно скажешь, танцует и солнышко улыбается, а если неправильно, то они грустные. Еще есть такая программа, я задумываю какое-нибудь слово про животных или про овощи, а компьютер должен угадать. Ну, он вопросы задает, а я должен «да» или «нет» отвечать. Я однажды специально неправильно ответил. Так компьютер потом сказал, что я нечестно играю. «Орган» есть. Папа сам эту про-

грамму написал. Показать? Только я музыку играть не умею».

Покинем Сашу с его компьютером, но не потому, что он «музыку играть» не умеет. Этому он, наверное, научится. Впрочем, компьютер сам может исполнять «музыку»: достаточно, имея перед глазами ноты, набрать несложную программу, а если у вас есть опыт, вы сможете сделать, чтобы эта музыка была не слишком однообразной. Типичные для домашних компьютеров встроенные синтезаторы звука позволяют имитировать трехголосное звучание практически любого музыкального инструмента в пределах восьми октав.

В школе введен новый курс «Основы информатики и вычислительной техники». Трудно за короткий срок оборудовать все школы персональными компьютерами. Но они там будут. Сначала по нескольку штук на школу, затем — целый компьютерный класс. Для этих ПК нужны будут программы. Очень много интересных, умных, увлекательных программ, облегчающих освоение компьютера, общение с ним, облегчающих написание других программ. Процесс этот можно начать, но его нельзя закончить, так как вычислительная техника быстро и непрерывно совершенствуется, и программы тоже должны совершенствоваться. Мне кажется, что ключ успеха в том, чтобы уже сейчас создавать такие условия и такие организационные формы, при которых огромная масса наших программистов в короткий срок смогла бы освоить новые для них виды программ: детские, школьные, домашние. Чтобы это были программы мирового уровня. Здесь очень подошел бы опыт новосибирской посреднической фирмы «Факел». Школьная и детская информатика — наилучшее для этого поле деятельности.

Скоро наши малыши, приходя из школы домой, будут делать домашние задания «по программированию», листать книжки, журналы и справочники с удивительными названиями: «Машинная графика», «Языки программирования», «Операционные системы», «Микропроцессорные средства и системы» и т. п. Им скоро потребуется компьютер дома. Ну а взрослые, что они будут делать дома

с компьютером? Ведь не все же работают программистами.

Пусть это покажется странным, но не программисты — основные потребители домашних ЭВМ (говорят же, что работники кондитерских фабрик дома конфет не едят). Проведенные в различных странах исследования того, кто и какие программы приобретает для домашних компьютеров, показывают, что на первом месте стоят игры и другие «программы домашнего досуга», на втором — «системы обработки текста», на третьем — «все остальное».

Существует довольно распространенное мнение, что компьютерные игры — это бездумное нажатие на клавиши, ничего не дающее для ума и души. Один мой знакомый, человек с большим педагогическим опытом, как-то сказал в разговоре: если он купит компьютер, то позаботится о том, чтобы игр в доме не было. Но виденное мной в семьях, где есть домашние компьютеры, убеждает меня в обратном. Дети довольно быстро «проскакивают» период запойного увлечения играми. Сошлюсь на пример моего знакомого шестиклассника Тимура. Когда его отец (по профессии — переводчик) привез из зарубежной поездки для своих переводческих нужд домашний компьютер, то первое время мать мальчика серьезно опасалась, что к чрезмерному увлечению сына телевизором добавятся игры на компьютере: все эти «автогонки», «бокс», «каратэ», «звездные войны», «невозможные миссии», «битвы за побережье» и прочее. И правда, первый месяц так примерно и было. Затем Тимур стал осваивать «Имитатор полета». Для того чтобы научиться взлетать, совершать полеты, а тем более приземляться, потребовалось освоить несколько сот страниц руководства к этой игре, в котором, между прочим, написано и такое: «Просим учесть, что программа ИМИТАТОР ПОЛЕТА не учитывает эффект турбулентности ветра на крыльях самолета». Мальчику пришлось научиться ориентироваться по карте и по приборам, «выходить» к аэродрому по сигналам радиомаяка, прочим авиационным премудростям. Все это не назавешь «бездумным нажатием на клавиши». Через неко-

торое время Тимур попросил меня научить его программировать, чтобы «самому научиться делать такие программы». Я сказал ему, что эта программа — плод многомесячной работы большого коллектива талантливых программистов, но тем не менее заниматься мы начали. Сейчас он с большим удовольствием делает домашние задания по программированию: рисует на экране графики тригонометрических функций, вращающиеся фигуры, лабиринты и многое другое, телевизор стал смотреть реже, а оценки по математике и другим предметам улучшились.

Следующие по популярности — программы обработки текста (их часто называют «редакторы текста»). Все, кому приходится много писать, а в эту категорию попадают не только писатели и журналисты, но и научные работники, врачи, переводчики, композиторы (да, да! — писать ноты с помощью компьютера не менее удобно, чем статьи о компьютерах, — правда, для этого требуется специализированный редактор текста), легко поймут, почему системы обработки текстов популярны среди взрослых. Хороший «редактор текста» позволяет не только набирать текст, исправлять опечатки, заменять во всем тексте одно слово другим (например, имя Саша словами «шестиклассник Тимур»), менять формат страницы, шрифт, хранить все промежуточные и окончательные варианты вашей работы на диске или магнитной ленте, но и вставлять в текст рисунки и графики, набирать многоязычные тексты, перекраивать целые параграфы и главы, проверять правописание и исправлять орфографические ошибки, подсчитывать количество слов, знаков и выяснять, какое слово вы употребляете чаще всего. Кстати, системы обработки текстов — программы очень сложные и порой очень дорогие (примерно в пятую часть цены компьютера того типа, на который они рассчитаны). Но для домашнего употребления часто достаточно простого «редактора текста».

Среди остальных применений домашнего и персонального компьютера — обучение, базы данных, использование для коммуникационных целей.



Обучать, при наличии соответствующих программ, компьютер может многому: программированию, вязанию, географии, биологии, печатанию десятию пальцами «вслепую», игре по нотам, иностранному языку, навыкам вождения автомобиля по скользкой дороге... всего не перечислишь.

Программы, именуемые «базы данных», пригодятся тем, кто любит хорошо организованный архив, где можно хранить и список книг из вашей библиотеки, и библиографию к диссертации, и любимые афоризмы, и многое другое, причем не просто хранить, но максимально быстро найти любую интересующую вас информацию.

«Коммуникационные цели» — еще одна многообещающая сфера применения домашнего компьютера.

При наличии небольшого устройства, которое называется модемом, компьютер

подключается к телефону, после чего за несколько минут вы можете переслать вашу статью коллеге, или научному руководителю (понятно, что для этого у них тоже должны быть модем и компьютер), или прямо в издательство. Или, если вам не хватает вычислительной мощности вашего ПК, за определенную плату можно подсоединиться к большой ЭВМ. Или запросить из библиотеки интересующую вас книгу, статью. Все это тоже будет. Пусть позже, чем «редактор текста» и «имитатор полета», но будет.

Может быть, Саша, Тимур и их сверстники не станут профессиональными программистами, но компьютер несомненно займет в их повседневной жизни достойное место. Как заняла много веков назад книга. Как заняли телефон и радио.

Компьютер станет повседневным инструментом. Инструментом знания.

### Компьютер помогает обувщикам

На Ленинградском производственном объединении «Скорострой» внедрен оригинальный комплекс, автоматизирующий разработку новых колодок для обуви. В него входят электронный прибор их обмера, ЭВМ ЕС-1022, графопостроитель. Данные прибора поступают в компьютер, который с помощью графопостроителя вычерчивает предлагаемую геометрическую развертку кожи, идущей на изготовление обуви. В результате оптимального раскроя заготовки экономится ценное сырье.

Данные, полученные от прибора обмера колодок, могут быть использованы и для составления программы их изготовления на станках с ЧПУ. Кроме того, ЭВМ поможет рассчитать размеры пресс-форм, необходимые для производства колодок.

Внедрение нового автоматизированного комплекса на «Скорострой» сократило затраты труда на разработку и изготовление обувных колодок, улучшило их качество, позволило чаще менять модели обуви, выпускаемой объединением «Скорострой». Опыт показал, что такой комплекс может успешно использоваться в системах автоматического проектирования на обувных предприятиях страны.

# ЭВМ в народном хозяйстве

В. ПОЛЯКОВ

## На суше и на море

Морское транспортное судно 80-х гг. этого столетия, пожалуй, отличается от своего собрата, построенного 40 лет назад, в большей степени, чем этот последний от парусника середины прошлого века. Механические двигатели избавили моряка от необходимости следовать капризу ветров и течений, но способы управлять судном, его различными агрегатами и механизмами зависели, как и в парусном флоте, от умения и выносливости членов экипажа. Количество матросов, исполняющих различные работы в рейсах, сократилось, и вместо 100, а то и 200 человек, входивших в состав экипажа парусника XVIII—XIX вв., на борту грузового парохода начала XX в. трудилось 40—50 человек, большая часть которых — кочегары — обслуживала топку ненасытных котлов паровой установки, приводящей в движение винт судна. Как и сотни лет назад, в навигационной рубке судна неотрывно следили за горизонтом 2—3 пары глаз, выискивая в туманных просторах океанских путей встречные пароходы или вспышку маяка на опасном рифе. Через каждый час или два менялись на вахте рулевые, даже мускулистые руки крепких матросов уставали крутить тяжелое колесо штурвала, который через примитивный механизм тросовых тяг поворачивал перо руля парохода.

Системы и приборы автоматического

управления различными операциями пришли на морской флот так же закономерно, как они появились на иных видах транспорта. Причиной активного внедрения автоматических и электронных регуляторов на торговом флоте послужило стремление максимально сократить экипаж, поскольку рационализация морского судоходства и морской торговли истребили дух романтики в морской профессии и в этой древнейшей отрасли человеческой деятельности наметился дефицит рабочих рук. В наши дни на борту самых больших танкеров и сухогрузов работают не более 25 человек, и эта тенденция сохраняется — уже есть суда океанского плавания с экипажем 15—17 человек.

На первый взгляд это нереально — провести со столь малочисленным экипажем 100-тысячтонное судно через океаны и попасть на погрузку в порт, окруженный опасными отмелями, где к тому же под влиянием притяжения Луны постоянно меняются течения и глубины. Однако сомнения исчезнут при знакомстве с современным судном. Все палубы и отсеки судов густо начинены автоматическими электронными устройствами, управляемыми от бортового компьютера. Микроэлектронная техника решает на флоте многочисленные задачи, которые могут быть условно сведены в три основные группы.

ЧТО  
МОЖЕТ  
ЭВМ



Практически все функции морского штурмана выполняет в наши дни судовой компьютер. Хитроумная задача определения места судна на поверхности моря, которая веками составляла одну из «тайн» штурманской профессии, решается в считанные секунды на основе простейшей программы. При этом возможности ЭВМ сдерживаются несовершенством основного источника необходимой для расчетов информации — человека. Поскольку задача поиска места в открытом море решается астрономическими способами, в компьютер нужно ввести данные о высоте светил, определяемой с помощью различного типа секстансов, отсчеты хронометров в момент фиксации высоты каждой навигационной звезды и счислимые координаты судна. Эти величины в обычной практике мореходного искусства определялись штурманом вручную со всеми присущими субъекту ошибками. Но требования к точности конечного результата были столь невелики, что архаичные приемы мореходной астрономии одинаково устраивали судоводителей всех времен и флотов. В принципе мореходная астрономия со всеми формулами и эфемеридами светил заложена в память компьютера и уже не менее двух десятков лет реализуется в навигационных комплексах иностранных и отечественных марок. Однако морская навигация все более теряет свою традиционную связь с естественными светилами и делает ставку на искусственные «радиозвезды». Все новые суда,



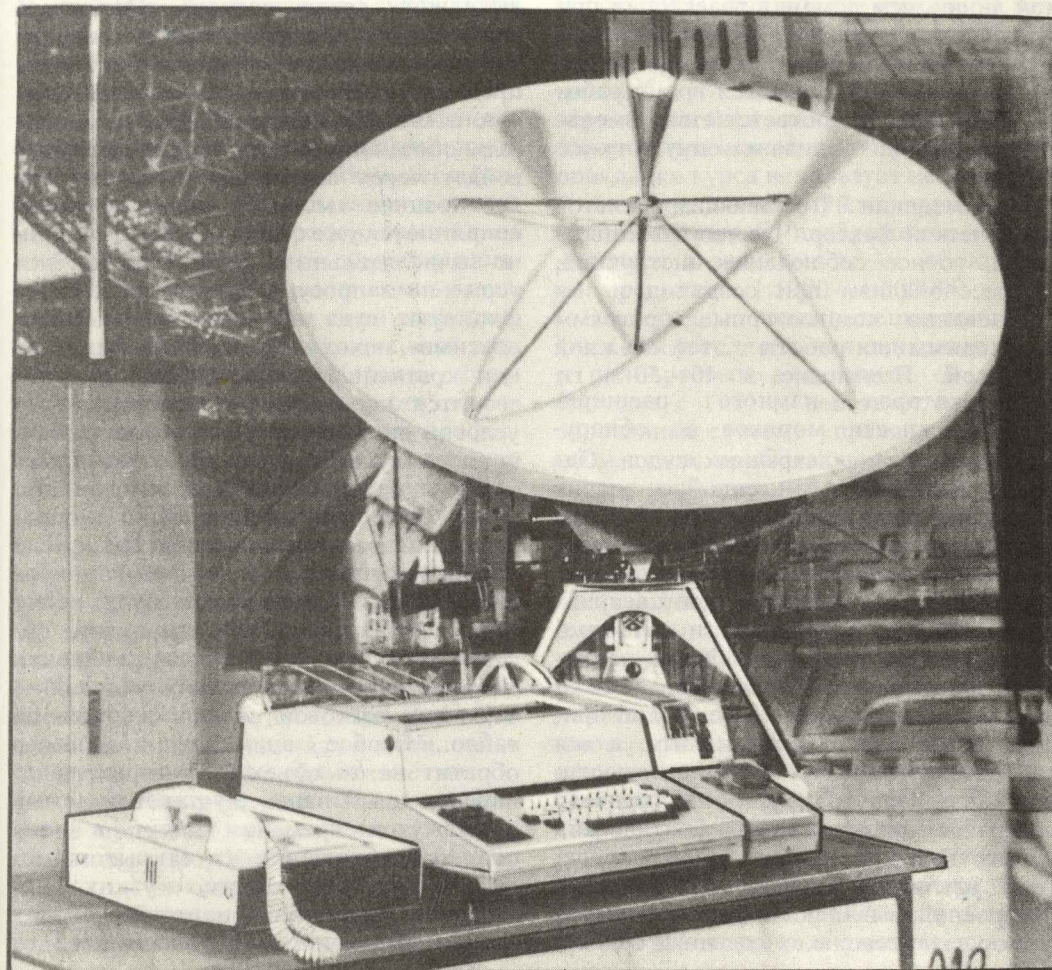
перевозящие грузы или выполняющие функции геологической или разведки, оборудуются приемоиндикаторами спутниковой навигации. Над Землей постоянно работают в активном режиме несколько цепочек морских спутников, находящихся на геостационарных орбитах с параметрами, позволяющими получить информацию практически круглосуточно и в любой точке океана. Приемоиндикаторы систем космической навигации настраиваются на рабочие частоты соответствующих групп навигационных спутников, принадлежащих различным государственным и частным организациям. Рабочий процесс определения позиции судна в море заключается в решении судовой ЭВМ параллактического треугольника, где опорными точками являются спутники. Достижения микропроцессорной техники позволили объединить радиоприемный комплекс и электронное решающее устройство в одном небольшом блоке, располагающемся непосредственно на рабочем столе штурмана. Точность космической навигации не идет ни в какое сравнение с традиционными способами, включая радионавигационные системы земного базирования. По мере совершенствования судовой приемно-вычислительной аппаратуры и выведения на орбиту дополнительных спутников с необходимыми параметрами точность данных все менее зависит от времени суток и наличия спутника на возвышенных углах сферы. Можно сказать, что система спутниковой навигации является в наши дни наиболее совершенной из всех имеющихся в распоряжении морского штурмана. Ее дисплеи, устанавливаемые на судах, выдают конечные результаты, которые являются сложным синтезом информации, получаемой в виде радиосигналов от спутников и постоянного потока данных от судового гирокомпаса и лага (указатели скорости и направления). Точность определения места, которую обеспечивают последние модели приемоиндикаторов спутниковых систем навигации, исчисляется в долях метра, что намного превышает практический запрос. Наряду с системами спутниковой навигации в судовых условиях успешно используются до сих пор наземные ра-

дионавигационные системы типа Декка, Лоран, Омега и прочие приемоиндикаторы. Все системы в основе своей являются простейшими электронно-вычислительными устройствами и на основе анализа радиосигналов двух излучающих станций, работающих в парных цепочках, дают конечный результат в виде параметров графических линий положения или в усовершенствованных моделях — в виде конечных географических координат.

Дальнейшее развитие микропроцессорной техники позволяет передать в руки электронно-вычислительных ма-

шин весь цикл управления судном. В этом случае на всем маршруте следования судна через океан заранее выбираются контрольные точки, в которых должна производиться корректировка курса или поворот на новый курс. Удержание судна на назначенном курсе производится обычным авторулевым, в котором электронные автоматы постоянно вводят поправки на ветровой или волновой снос. Сложнее учитывать влияние морских течений, результат воздействия которых может быть выявлен только при сравнении фактической позиции судна с контрольными точками на

Аппаратура «Стандарт-А» со встроенными микро-ЭВМ обеспечивает надежный прием радиосообщений судами





заранее определенном маршруте. До внедрения электроники в процесс судовождения этим занимался капитан судна, по нескольку раз в сутки задавая новый курс матросу-рулевому или автоматическому устройству, управляющему рулем судна. В новейших роботизированных навигационных комплексах эту работу выполняет кибернетический капитан. На электронных картах с заданной периодичностью производится сопоставление фактических и контрольных координат, и сигналы коррекции курса поступают на автоматы, управляющие рулем судна. При этом, помимо высвобождения человека от напряженной, но примитивной аналитической работы, достигается другой важный эффект — экономия топлива. Линии пути, по которому ведет судно кибернетический капитан, гораздо ближе к прямой линии, чем ломаная траектория при ручной навигации. Тем самым сокращается длина пути между точками отхода и назначения и экономится время работы двигателя и соответственно уменьшается расход топлива на одну и ту же работу.

В применении ВТ решающим является человеческий фактор. Чувство ответственности, точное соблюдение инструкций, профессионализм при создании и использовании компьютерных программ необходимы для работы с этой сложной техникой. Появление в 40—50-х гг. радиолокаторов намного расширило возможности моряков по обнаружению в тумане встречных судов. Однако обнаружить — еще не значит безопасно разойтись. И суда, оборудованные самыми совершенными для своего времени радиолокационными станциями, продолжали в тумане идти к аварийному сближению, навалам и катастрофическим столкновениям. Экран радиолокатора давал картину «относительного движения», поскольку «свое» судно принималось условно неподвижным, а вся картина окружающей обстановки постоянно двигалась, создавая у наблюдателя превратное представление о фактических курсах соседних судов. Были изобретены различные способы графических построений, выявляющих истинные курс и скорость нескольких соседних судов и

степень опасности, которую они представляли для наблюдателя, но... в проливе Ла-Манш на экране РЛС постоянно высвечиваются десятки судов и каждое из них через считанные минуты может оказаться опасным. Простейший компьютер, подключенный к радиолокатору, ввел в сигнал, формирующий изображение, поправку, учитывающую курс и скорость своего судна. Эта маленькая революция в радиолокации значительно изменила взаимоотношения судоводителя с этим сложным прибором. Картина на экране РЛС изменилась, она стала показывать фактическое соотношение курсов и скоростей, приобрела абсолютный характер. Теперь судно, на котором находится наблюдатель, уже не стояло на месте, а вместе со всеми соседними судами двигалось вдоль береговых очертаний, ничем не выделяясь среди соседей. Появилась возможность проще и нагляднее анализировать взаимные перемещения судов, предвидеть возможные изменения курсов, выявлять опасные суда и оценивать обстановку в районах, куда судно войдет через определенное время. Более поздние модели радиолокаторов сопрягаются уже со сложными электронно-вычислительными устройствами, которые по запросу наблюдателя классифицируют суда на «опасные» и «безопасные», исходя из параметра «дистанции кратчайшего сближения», которая вводится капитаном в зависимости от условий видимости. С помощью «электронного щупа» каждое судно, вошедшее в зону радиолокационной видимости, может быть идентифицировано и определены элементы его движения. Если в дальнейшем это судно меняет курс или скорость, наблюдатель будет извещен компьютером об изменении обстоятельств плавания. Форма извещения выбирается в зависимости от условий — звуковой, световой сигнал, надпись на табло. Особое внимание компьютер обратит на те объекты, которые через определенный, заранее устанавливаемый промежуток времени войдут в зону опасного сближения (в открытом море — 2 мили или более, в узких фарватерах, проливах, портовых водах допуск уменьшается в зависимости от

скорости судна и размеров акватории для маневрирования). Во избежание риска опасного сближения, компьютер заранее рассчитает оптимальный вариант маневра и покажет его на одном из табло. При этом компьютер обязательно учтет радиус и время поворота судна на новый курс (для крупных морских судов эти величины составляют сотни метров и десятки секунд), а по заданию может учесть и дополнительные факторы риска: наличие сноса от течения, ветра, прилива, проседание на мелководье. Расчетный маневр на расхождение может быть реализован в виде команды на исполнительный механизм — перо руля или регулятор оборотов двигателя. Фактическое завершение задачи расхождения на крупнотоннажных судах доверяется компьютеру уже сегодня в открытом море. В районах интенсивного судоходства решение ЭВМ подвергается в условиях нормальной видимости дополнительному анализу капитаном или лоцманом, что можно объяснить не объективной необходимостью, а скорее традиционной склонностью моряков «считать себя ближе к опасности». При плавании в тумане или в дождь, когда радиолокационная информация является по существу единственно надежной, выводы компьютера, основанные на анализе этой информации, принимаются капитаном или лоцманом к использованию. Моряки, у которых компьютерная грамотность закладывается с самого начала их профессионального обучения, вполне осознанно приходят к выводу, что в современных условиях ошибки самых опытных и предусмотрительных судоводителей имеют большую вероятность и обходятся дороже, чем последствия отказа электронно-вычислительного комплекса, сопряженного с РЛ-станцией.

Возможности компьютерных навигационных установок постоянно расширяются как в плане их надежности, так и по функциональным возможностям. До сих пор подавляющее число морских судов получает информацию о районе плавания с морских карт, достоинства и недостатки которых представляет себе каждый. С появлением в распоряжении судоводителя электронно-вычислитель-

ных средств координаты судна, полученные через систему спутниковой навигации, периодически фиксируются на перфоленте.

Для плавания вблизи берега судовой компьютер должен хранить в памяти кроме сетки географических координат массу информации, которую обычно капитан получает из морских карт и лоций. Морская карта прочитана на языке ЭВМ, и вся необходимая информация записана на носителях, которые включаются в работу по мере прохождения судном соответствующих районов плавания. Навигационная обстановка вводится компьютером на экран судовой РЛС в виде отметок глубин, подводных опасностей, очертаний берегов, силуэтов маяков и бакенов и синхронно сопрягается с фактическим изображением этого участка, получаемым при работе РЛС. Английские инженеры разработали модель видеопрокладчика курса, который дает изображение, раскрашенное в 7 цветов. Практически на экране дисплея капитан видит изображение привычной карты, но только для прокладки курса использует не карандаш и линейку, а систему электронных визиров. Масштаб изображения может меняться в зависимости от обстановки в пределах 1:5000—1:500000.

Электронные карты успешно применяются для регулирования движения судов в припортовых акваториях. В заливе Находка с 1981 г. действует автоматизированная система управления движением судов, прибывающих в порты Находка и Восточный. В этой системе электронная карта представляет собой крупномасштабную видеосхему всего залива, на которой выделены условными электронными символами судоходные фарватеры, зоны якорных стоянок, опасные зоны. На эту схему проецируются радиолокационные отметки судов, находящихся в заливе, и центральный компьютер системы определяет, нет ли у судов, движущихся по фарватерам или стоящих на рейде, опасного сноса в сторону мели. Если возникает риск коснуться грунта или столкнуться со встречными судами на фарватере, компьютер дает совмещенный свето-звуковой сигнал тревоги и указывает, какие





суда в опасности. Дежурный оператор станции управления движением предупреждает по радиотелефону капитанов или лоцманов соответствующих судов о возникновении опасности и дает рекомендации для ее устранения.

Электронно-вычислительная техника используется на морском флоте и для обучения будущих капитанов. Для этой цели во всех крупных портах СССР организованы морские радиолокационные тренажеры на базе автоматизированной навигационной системы «БРИЗ-Е». В тренировочных центрах курсанты морских училищ проходят полный цикл обучения, а капитаны и их помощники периодически осваивают новые методы компьютерного анализа условий плавания. Тренажеры построены по принципу моделирования критических ситуаций на морских дорогах и программируются опытными операторами с учетом различных условий плавания и типов судов.

#### Обеспечение оптимальной загрузки судов, слежение за грузами, подготовка транспортной документации.

Вопрос, как разместить грузы в трюмах, в наши дни решается с помощью автоматики. Еще до подхода сухогруза или танкера к причалу ЭВМ порта делает предварительный обсчет карго-плана (грузового плана) для конкретного судна с учетом всех характеристик грузов, размеров грузовых помещений на судне, требований сохранения прочности

его корпуса в самых экстремальных условиях и предстоящей последовательности выгрузки в различных портах. На нефтеналивном танкере такой план после его проверки на судовой ЭВМ и утверждении капитаном становится программой для автоматизированного процесса загрузки. Судовая ЭВМ используется для управления грузовыми операциями и дает команду открыть клапаны на трубопроводах, по которым жидкий груз начинает поступать в отсеки танкера из береговых накопительных емкостей. По мере заполнения грузовых танков сигнальные датчики сообщают объем и температуру принимаемого груза. Эта информация обрабатывается на ЭВМ с учетом нормативов остойчивости и прочности корпуса танкера. В нужный момент компьютер дает команду о переключении потока груза в другой танк, а затем и о полном прекращении погрузки. К этому времени вычислительное устройство судна печатает на телексом аппарате отчетный документ. Подобная информационно-управляющая система широко используется на ряде крупнотоннажных танкеров и балкеров, что позволило сократить обслуживающий персонал и повысить надежность определения количества груза и его правильного размещения в отсеках танкера.

Для судов-контейнеровозов ЭВМ производит аналогичный расчет, принимая во внимание различный вес сотен и даже тысяч стандартных контейнеров, которые необходимо перевести в различные порты и для различных получателей. Затраты ручного труда на такую работу составили бы много дней для целой группы операторов. Электроника выдает готовые карто-планы за считанные часы, а затем в течение рейса производит их регулярную корректировку для оптимального распределения контейнеров на борту судна по мере их выгрузки в портах и приеме порожних на борт.

Незаменима роль вычислительной техники для современных контейнерных терминалов в портах. При одновременном хранении на площадках терминала десятков тысяч груженых и порожних контейнеров только быстродействующему электронно-вычислительному устрой-

ству под силу произвести поиск партии контейнеров, предназначенных для погрузки на определенное судно, и определить порядок их подвоза для укладки на контейнеровоз.

Если контейнер при перевозке или погрузке получает повреждение, информация об этом вводится в память системы, которая найдет владельца контейнера и сообщит его представителю, что необходим ремонт.

#### Автоматизация контроля за работой судовых механизмов и диагностика их неисправностей.

По мере сокращения штатов в судовых экипажах происходило все большее проникновение электронно-решающих устройств в машинные отделения судов. На автоматизированных судах по команде электронного оператора производится периодический контроль десятков и со-

тен параметров работы многих механизмов и агрегатов. При отклонении от нормы дается сигнал в помещение, где находится вахтенный персонал судна, с указанием характера неисправности. Если дефект продолжает развиваться, следящая электронная система производит автоматическое переключение неисправного механизма на дублирующий, а при необходимости отключает ряд агрегатов, нормальная работа которых невозможна без вышедшего из строя механизма или прибора.

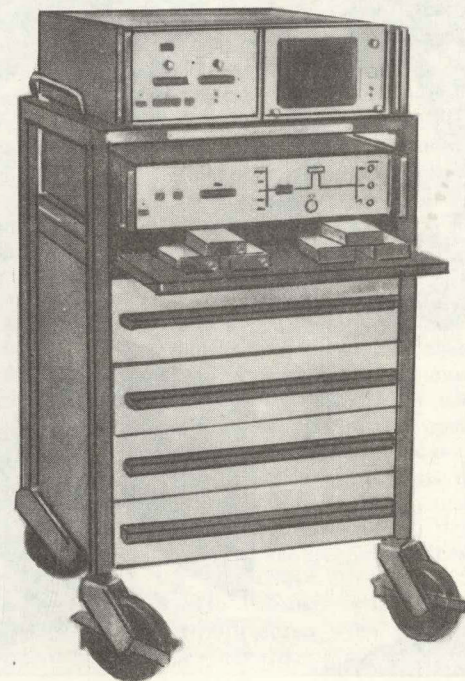
Созданы электронные системы диагностики судовых двигателей. В программу их работы заложены рекомендации заводов, которые строят и монтируют на судне те или иные механизмы и агрегаты. Такие системы значительно расширяют возможность оперативного поиска причин отказов и устранения неисправностей в судовых условиях.

#### Электроника — врачам

Сейчас электроника помогает врачам лечить многие недуги. Ее возможности расширяются благодаря использованию микропроцессоров в различной медицинской аппаратуре. Их применение в медицинской технике позволяет точно поставить диагноз, следить за состоянием пациента в больничной палате, вести терапевтическое лечение.

У нас в стране уже созданы компьютерные устройства, исследующие сердечно-сосудистую систему. Среди них радиокордиоанализатор РКАЗ-01 (см. рис). С его помощью врачи могут изучать кровообращение больного, контролируя его состояние. Такой анализатор автоматически измеряет объем циркулирующей в организме крови, читает кардиограммы. Данные анализа, графики, цифры выводятся на экран медицинского монитора и с помощью графопостроителя переносятся на бумагу.

Применение радиокордиоанализаторов в клиниках повысит эффективность лечения сердечно-сосудистых заболеваний, поможет врачам в борьбе с этими недугами.



ЧТО?  
МОЖЕТ  
ЭВМ





РЕПОРТАЖ  
НОМЕРА

## Латвийский опыт: вуз помогает школе

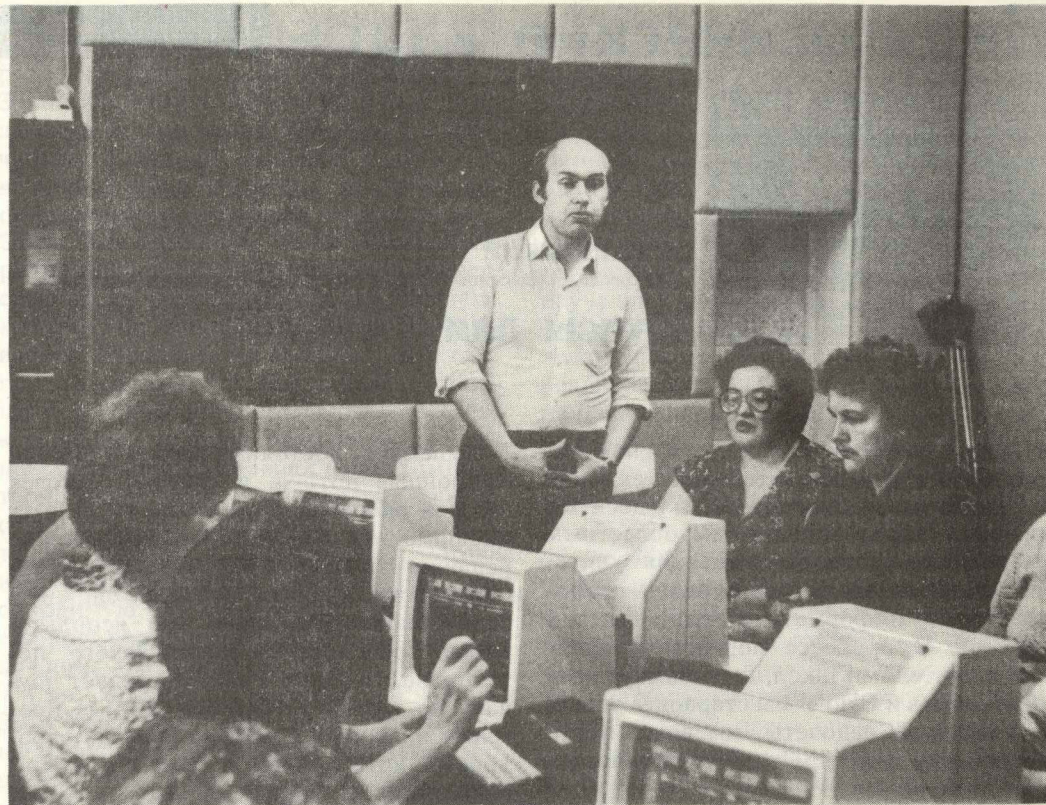
— Знакомьтесь, это наша лаборатория проблем школьной информатики, — Русинь Висвалдович Фрейвалд, заместитель директора Вычислительного центра Латвийского государственного университета им. П. Стучки, приглашает нас в одну из университетских аудиторий. — Как видите, здесь есть разнообразная вычислительная техника для внеклассной работы школьников и подготовки педагогов. В этом реальная помощь вузовских специалистов учителям информатики.



В прошлом году к нам поступили компьютеры, на базе которых в лаборатории созданы 10 ученических, 1 учительское и 5 инструментальных рабочих мест. Эта техника к нам поступила в мае — июне, а уже 1 июля стала действовать локальная сеть. Подготовлены латышские и русские версии комплекса обучающих программ. Это дало возможность эффективно использовать их на курсах переподготовки учителей. Кроме того, специалисты лаборатории разработали



2



109

3

автоматизированную библиотеку для хранения программ. Проведены успешные эксперименты обмена информацией между компьютерами по телефонным каналам и подключению рабочих мест к компьютеру СМ-4, создана методика обучения детей основам программирования. Так решаются актуальные задачи применения ВТ в учебном процессе.

Лаборатория проблем школьной информатики в ЛГУ действует всего лишь два года, но о ее работе известно далеко за пределами республики. Ученые лаборатории установили деловые контакты с ведущими научно-исследовательскими институтами Академии педагогических наук СССР, многими лабораториями школьной информатики.

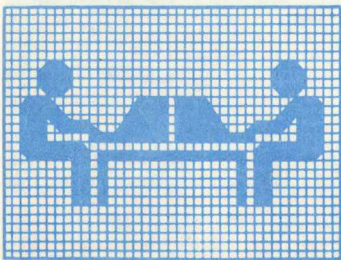
Частые гости в ЛГУ — рижские школьники, которые под руководством специалистов посещают внеклассные занятия. Ребята осваивают компьютеры (фото 1), учатся основам программирования. Например, 8 учащихся школы № 73 в прошлом году получили аттестат зрелости с записью о том, что они прошли факультативный курс при лаборатории проблем школьной информатики.

Она стала своеобразным «полигоном»

для испытаний школьных компьютеров. По соглашению с Минпросом Латвийской ССР вычислительная техника, поступающая в школы республики, будет проверяться в лаборатории, а учителя получают здесь навыки работы. Это лишь часть реальной помощи ученых университета педагогам. Для них организованы курсы подготовки по основам информатики, разработаны 20 демонстрационных программ, рассчитанных на первое знакомство учителей с компьютером. Проводились и практические занятия по изучению языка Бейсик. В помощь педагогам подготовлено 16 консультантов, которые на летних курсах читали лекции по основам информатики и вычислительной техники, помогали учителям в изучении программирования на микро-ЭВМ и микрокалькуляторах. Для педагогов были организованы целенаправленные практические занятия на базе школьных кабинетов вычислительной техники (фото 2, 3) и в университете, где специально для работы с педагогами и учащимися созданы дисплейные классы на базе ЭВМ СМ-4.

По отзывам учителей, такая форма сотрудничества вуза и школы даст хорошие результаты.





М. РАТИНСКИЙ

Московский областной институт усовершенствования учителей

## Курсы для учителей

В новом учебном году курс «Основ информатики и вычислительной техники» изучается как в девятых, так и в десятых классах. Перед Московским областным институтом усовершенствования учителей стояла задача подготовить преподавателей информатики для десятых классов. В процессе подготовки занятий с учителями школ, преподавателями техникумов и СПТУ мы старались учесть и устранить недочеты, имевшие место в прошлом учебном году. Остановимся на них.

Методологическая ошибка, допущенная при определении наименования предмета, смешала воедино как цель, т. е. то, чему следует обучать в школе, так и средство для достижения цели. Поэтому вопрос, идентично ли обучение информатике обучению программированию, своей актуальности так и не потерял.

Одна из основных задач совершенствования образования — вооружение учащихся знаниями и навыками использования современной вычислительной техники. Как расшифровать эту задачу? Судя по рекомендованным для школы пробным учебным пособиям и программам, их авторы понимают ее следующим образом: дать всем учащимся сведения по устройству ЭВМ и некоторые навыки по программированию. Но правилен ли этот путь?

Для успешного процесса обучения необходимо умение грамотно работать с различного рода информацией (гуманитарной, естественнонаучной, технической). С каждым годом объем такой информации непрерывно увеличивается.

Больше того, постоянно растут и информационные потоки, в которых человек должен оперативно выбирать именно ту информацию, без которой невозможно решение стоящих перед ним задач. Действительно, решить быстро сложную задачу возможно лишь при использовании современного технического потенциала. Поэтому в школе следует обучать не только устройству ЭВМ и программированию; в первую очередь необходимо дать знания о том, какую информацию и каким образом следует обрабатывать на ЭВМ, т. е. осуществлять подготовку всех школьников как потребителей ЭВМ. Исходя из этих соображений в Московском областном институте усовершенствования учителей и проводилось обучение на летних курсах.

В период с 15 по 31 июля 1986 г. в МОИУУ была организована подготовка учителей информатики (в ней принимали участие и другие вузы: МФТИ, МЛТИ, МОПИ, Коломенский и Орехово-Зуевский пединституты). Во всех пунктах занятия проходили по единой методике и программе. Перед началом занятий в МОИУУ для преподавателей был проведен установочный семинар.

Прошли курсовую подготовку 3500 учителей математики, физики, химии. Они занимались по 36-часовой программе, преподаватели других предметов — по 12-часовой.

Обучение всех категорий учителей проводилось исходя из следующих основных положений. Во-первых, была выделена цель, т. е. сама основа предмета, а именно информатика — наука о структуре и общих свойствах ин-

формации. Во-вторых, определено наиболее оперативное средство для решения ряда важнейших задач информатики — ЭВМ. Поэтому в зависимости от отведенного на подготовку времени все обучаемые получили сведения о трех основных базовых понятиях школьной информатики: информация — алгоритм — ЭВМ. Это позволило более эффективно разъяснить на курсах вопрос непосредственно о компьютерной грамотности, которая, по нашему мнению, обеспечивается изучением не только основ программирования и ознакомления с принципами работы на ЭВМ, но и всего комплекса учебных предметов школьной программы. Главная цель компьютерного обучения — воспитание творческой личности, свободной от рутинного труда, умеющей из огромного потока информации выбрать действительно необходимую и достаточную, способной в единицу времени добиваться больших и лучших результатов.

При обучении учителей не менее 50 % времени было отведено работе на ЭВМ. Для преподавателей, занимающихся по 12-часовой программе, эта работа включала следующие элементы: ознакомление с устройством ПЭВМ, освоение клавиатуры, ознакомление с пакетом прикладных программ, компьютерный урок. При проведении компьютерного урока, в зависимости от профиля обучаемых, использовались программы по русскому языку, литературе, биологии, химии, физике, математике.

Учителя, проходившие подготовку по 36-часовой программе, были, кроме того, ознакомлены с основными командами языка Бейсик, и каждый из них самостоятельно составил 2—3 элементарные программы по предмету.

При проведении практических работ учителя информатики самостоятельно разрабатывали алгоритмы, решали задачи по следующей схеме: составление блок-схемы алгоритма — описание блок-схемы посредством алгоритмического языка, приведенного в пробном учебном пособии. Затем по разработанным алгоритмам сами учителя составляли программы. В качестве языка программирования использовался реальный язык используемой на занятиях техники. Сле-

дует отметить, что при отработке темы «Языки программирования» основное внимание было уделено общей структуре построения языка, а не бессмысленным (в связи с отсутствием широкой элементной базы) рассуждениям о преимуществе одного языка программирования над другим. К сожалению, эти рассуждения зачастую уводят от основного вопроса — чему следует обучать при изучении информатики.

Нами был проведен эксперимент. Учителям, после того как они практически реализовали на ПЭВМ написанные ими на Бейсике программы, были даны основные элементы языка Фокал и было предложено переписать программы на этом языке. Задание не вызвало никаких затруднений. Таким образом, учителя поняли, что, зная общие принципы программирования, можно довольно легко адаптироваться к реальному языку элементной базы и совершенно незачем все обучение по предмету сводить к обучению программированию вообще (чему отдается 90 % всего времени, отведенного на предмет в школе), а тем более программированию на одном конкретном языке.

В процессе обучения в МОИУУ учителя поняли, что они не являются преподавателями программирования, а должны учить принципам и методам правильной работы с информацией любого рода. Поэтому у нас не стоит остро вопрос о так называемом безмашинном варианте обучения. Да, обучение программированию в безмашинном варианте столь же бессмысленно, как и обучение игре на музыкальных инструментах при отсутствии последних. При обучении же непосредственно информатике как таковой компьютер на первых порах не нужен.

Школам необходима не любая вычислительная техника, а дисплейные классы, где ПЭВМ объединены локальной сетью. Только в таком классе возможен целенаправленный диалог учителя с учениками.

Исходя из возможностей оснащения школ техникой в настоящее время обучение по предмету целесообразно сосредоточить в межшкольных кабинетах информатики. Кроме того, обучение не-



обходимо сконцентрировать по времени. В межшкольном кабинете можно эффективно организовать обучение учащихся шести школ — по одному дню в неделю для каждой.

По нашему мнению, обучение в этом случае следует осуществлять по следующей схеме:

IX класс — 2 ч в неделю во втором полугодии (часы спаренные);

X класс — 4 ч в неделю в первом полугодии (занятия в один день).

В IX классе второе полугодие отведено для подготовки к экзаменам и ни по одному предмету практически не дается новый материал. Поэтому нет необходимости делать исключение для столь сложного предмета, как информатика, который рассматривается как один из фундаментальных компонентов всеобщего среднего образования. К тому же на занятиях по информатике учащиеся уже сейчас вполне могут совершенствовать свои знания по ряду разделов математики, физики, биологии, химии, а в недалеком будущем и других предметов.

Подобная постановка обучения позволит дать учащимся многие необходимые умения: определять (формулировать) задачу; находить пути ее решения (разрабатывать математическую модель); разрабатывать алгоритм решения и, при необходимости, осуществлять программирование на конкретной ЭВМ.

Курсовая подготовка учителей информатики завершалась итоговой конференцией, на которой слушатели имели возможность высказать свое мнение по таким важным вопросам, как организация и качество проведения занятий на курсах, приемлемость для школы рекомендованных учебных пособий и организация занятий по предмету в школе.

Кроме того, около 1000 слушателей ответило на вопросы анкеты, предложенной им. Приведем обобщенные ответы, отражающие мнение более 90 % опрошенных.

— Каково ваше отношение к введению в программу школы нового предмета?

— Весьма положительное. Необходимо было это сделать значительно раньше

и изыскать возможность проведения практических занятий с использованием ЭВТ во всех школах.

— Чему следует обучать на уроках информатики?

— Структурно-логическому мышлению во всех сферах деятельности.

— Кто должен вести уроки по информатике?

— Учитель-предметник, а не учитель-совместитель, как это происходит сейчас.

— Ваше мнение об организации и проведении курсовой подготовки.

— Занятия организованы и проведены хорошо. Необходимо значительно увеличить время на курсовую подготовку.

— Какой вам представляется роль информатики в общей системе школьного обучения?

— Правильное обучение информатике позволит осуществлять реальную межпредметную связь на всех этапах.

— Как вы оцениваете пробное учебное пособие?

— Это не пособие по информатике, поскольку совершенно не рассмотрено одно из основных базовых понятий предмета «информация». Пособие написано языком, неприемлемым для массовой школы, и изобилует серьезными ошибками.

И в заключение отметим следующее.

Порой высказываются суждения о том, что информатика в недалеком будущем сольется с математикой. Подобные заявления делают и весьма уважаемые в науке люди.

Что это — стремление таким путем спрятать методологический просчет, допущенный при определении содержания, так и наименования нового школьного предмета? Но подобное «исправление» допущенной ошибки, кроме вреда, ничего не принесет. Надо называть вещи своими именами. В последнее время появились новые науки — биохимия и биофизика. Но биология не сделала после этого своими разделами ни химию, ни физику.

Хотелось бы, чтобы высказывающие подобные идеи помнили о таких свойствах информации, как определенность и достоверность, которых явно недостает в пробных учебных пособиях, которые многие упорно именуют учебниками.

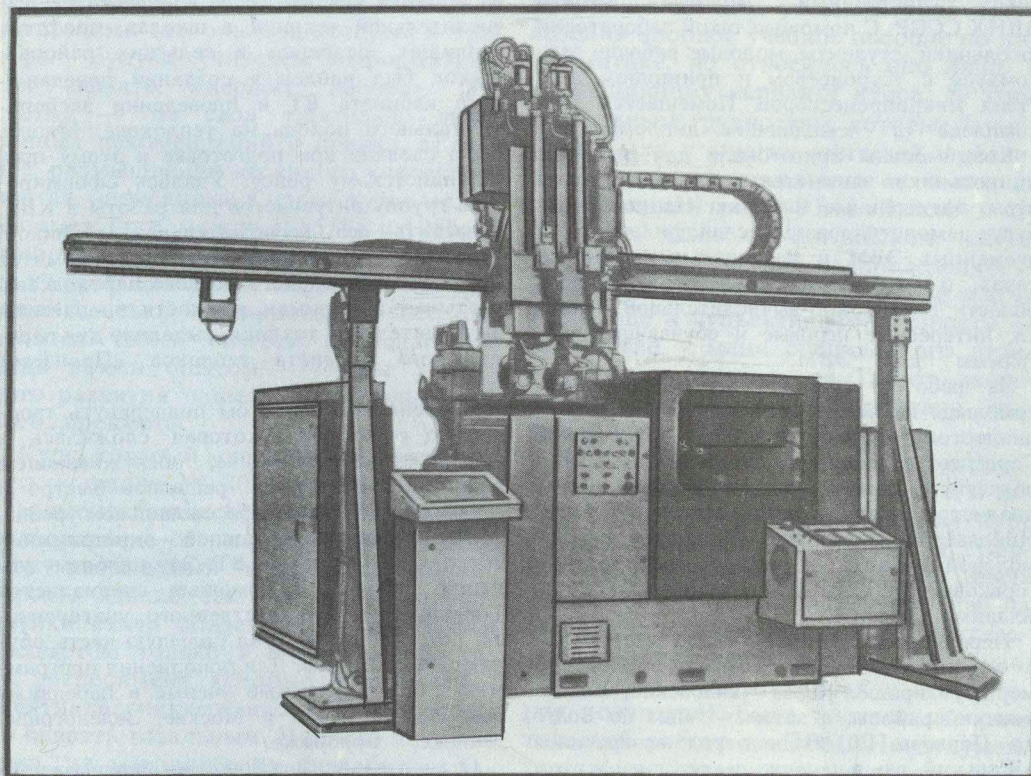
# ЧТО? МОЖЕТ ЭВМ

## Комплекс с ЧПУ

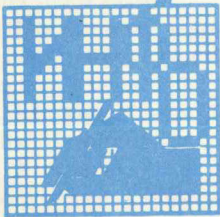
Советские станкостроители разработали оригинальный комплекс, выпускающий детали типа фланец. Управляет таким комплексом система числового программного управления «Электроника НЦ-31».

В состав комплекса (см. рис.) входят токарно-револьверный станок с ЧПУ, промышленный робот и дополнительные устройства для хранения и транспортировки заготовок и готовых деталей.

Производительность нового комплекса 17 деталей в час. Вычислительная техника автоматизировала процесс металлообработки, повысила производительность оборудования. Улучшилось качество продукции, сократились расходы сырья.







**Ю. ТУТАРИНОВ**

Горьковский областной совет молодых ученых и специалистов

**А. ЧИНЯЕВ**

Горьковский обком ВЛКСМ

## «Пропагандист»

Минувшим летом из Горького вверх по Волге отправился агиттеплоход «Пропагандист». На его борту — экспериментальный передвижной обучающий вычислительный класс (ЭПОВК), созданный по инициативе областного совета молодых ученых и специалистов под руководством Горьковского обкома комсомола. Основная цель этого рейса — познакомить школьников, молодежь с возможностями вычислительной техники, научить их работать с ЭВМ. На теплоходе установлены компьютеры ДВК-2М, «Электроника-60, ДЗ-38». Есть в ЭПОВК, своеобразном кабинете ВТ, и уникальная новинка: микропроцессорный комплекс, удостоенный Золотой медали ВДНХ СССР. С помощью такой лаборатории школьники, студенты, молодые рабочие знакомятся с устройством и принципом действия микропроцессоров. Помещается этот комплекс в чемоданчике-«дипломат».

Комсомолы приготовили для пионеров и школьников занимательные компьютерные игры, электронные игрушки. На теплоходе будут демонстрироваться слайдфильмы о современных ЭВМ и персональных компьютерах, о достижениях советской науки в области электронно-вычислительной техники, интересные игровые и обучающие программы для ЭВМ.

За работой кабинета ВТ следят комсомолы из числа молодых ученых и специалистов предприятий и учебных заведений Горького. Руководит коллективом член областного совета молодых ученых и специалистов, кандидат технических наук Юрий Николаевич Тутаринов. В составе отряда молодой педагог — учитель информатики горьковской школы № 174 Чиняева Людмила Владимировна.

Передвижной кабинет сначала остановился в пионерском лагере «Сокол», далее его маршрут прошел через Чкаловский, Болонинский районы, а затем — вниз по Волге до Павлова. 20 июля отряд возвратился в Горький, где в течение десяти дней в этом

кабинете стали обучаться работники аппарата обкома, городских и районных комитетов комсомола Горьковской области.

Подводя итоги работы кабинета ВТ на теплоходе «Пропагандист», нужно отметить, что его поход удался. Об этом свидетельствует большой интерес, который проявили к передвижному кабинету ВТ свыше 2 тыс. посетителей. Об этом же говорят восторженные отзывы школьников, учащихся ПТУ, преподавателей, хозяйственных и комсомольских работников.

Идея создания такого кабинета родилась в областном комитете комсомола в начале 1986 года. Пришли мы к ней, чтобы каким-то образом компенсировать недостаток вычислительной техники в школах, профтехучилищах, особенно в сельских районах. Выход был найден в создании передвижного кабинета ВТ и проведении экспериментального похода на теплоходе. Многие было сделано при подготовке к этому пропагандистскому рейсу. Удалось сформировать группу энтузиастов для работы в КВТ, оснастить его вычислительной техникой.

Руководство, партком и бассейный комитет профсоюза Волжского речного пароходства, учитывая и понимая важность пропаганды вычислительной техники, выделило для передвижного кабинета теплоход «Пропагандист».

Особенно хотелось бы подчеркнуть творческую атмосферу, которая сложилась в молодежном коллективе, обслуживающем кабинет. Все вопросы решались быстро и четко, на высоком профессиональном уровне. А их было немало. Основной — программное обеспечение кабинета ВТ. Эту проблему успешно решили с помощью специалистов Горьковского государственного университета, который разработал большую часть обучающих программ. Для пополнения программного фонда молодые ученые в передвижном ВТ побывали в Москве, Зеленограде, Минске, Воронеже.

Отличительная особенность передвижного

кабинета — его универсальность. Он может использоваться для обучения школьников, учителей, преподавателей ПТУ, партийных, комсомольских, хозяйственных работников. Для каждой категории слушателей свои лекции, практические работы, демонстрационные программы. Для обучающихся в передвижном кабинете всегда найдутся специальные программы, интересные каждому.

Огромной популярностью пользуются у детей, да и у взрослых электронные игрушки с программным управлением. Например, «Луноход», игрушка «Ну, погоди!», электронный тир «Квант» и др. Для учителей информатики и специалистов в области вычислительной техники на борту теплохода была развернута выставка методической и специальной литературы. Во время работы кабинета проводились также социологические исследования.

**Д. СЛАВИН**

## Каждый урок — эксперимент

Вот и закончился учебный год. Выставлены годовые оценки, есть время сесть за письменный стол и поделиться с читателями тем опытом, который я приобрел, будучи учителем информатики, так сказать, «первого набора». Эта статья — не свод тщательно выверенных рекомендаций и правил, а, скорее, размышления «военного корреспондента», написанные в промежутке между «боями» (т. е. между 1985/86 и 1986/87 учебными годами). Я буду рад, если кто-то из преподавателей информатики, прочитав статью, откликнется на нее, поделится своим опытом преподавания. Ведь обмен опытом — залог успешного развития нашего, еще столь молодого, предмета.

Курс, который мне пришлось преподавать в школе № 204 Москвы, был экспериментальным, т. е. рассчитанным на преподавание с использованием ЭВМ (так называемый машинный вариант курса). Однако, так как основы информатики стали обязательным школьным предметом только в этом учебном году, эксперимент, который проводился с девятиклассниками школы № 204, можно назвать тотальным. Не было ни учебника, ни методических разработок, да-

Итак, закончен первый этап работы передвижного кабинета ВТ. Что же дальше?

Сейчас обсуждается идея создания в области постоянно действующего передвижного вычислительного кабинета, который мог бы решать не только пропагандистские задачи, но и помочь в обучении школьников основам информатики и вычислительной техники.

Интерес представляет и организация молодежного компьютерного клуба, где был бы региональный банк обучающих программ. Можно в крупных районных центрах организовать небольшие передвижные кабинеты ВТ, актуальность и возможность создания которых подтверждается опытом горьковчан.

Ближайшая наша задача — расширение передвижного кабинета, создание на его базе областного компьютерного клуба.

же содержание курса к началу занятий не было до конца сформулировано. Правда, к началу учебного года было создано пробное учебное пособие по информатике, но относилось оно лишь к безмашинному варианту курса. К этим глобальным трудностям, которые, думается, были у многих учителей, добавились трудности локальные — новая, не опробованная микро-ЭВМ «Ямаха» и отсутствие, даже на английском языке, документации по ней; неопределенность сроков установки КУВТ в школе и т. д. Была еще и субъективная трудность — отсутствие у меня специального педагогического образования. Так что эксперимент получился действительно «тотальным».

Класс «Ямах» был введен в эксплуатацию лишь в январе 1986 г., т. е. в начале второго полугодия. Поэтому курс можно разделить на две части. Первая из них (первое полугодие) была чисто теоретической, упор делался на объяснение ученикам фундаментальных понятий типа систем счисления, алгоритмов и т. д. Вторая часть (второе полугодие) была посвящена практической работе на ЭВМ. Не подумайте, что я отношусь к разряду людей,



которые считают, что задача нового курса — научить школьников составлять алгоритмы и программы. Это не так. Я — сторонник более широкой трактовки содержания курса, по которой школьников помимо основ программирования следует научить методам формализации задач, основам классификации, работам с базами данных, электронными таблицами, редакторами текстов и т. д. Однако такое содержание курса должно поддерживаться адекватным программным обеспечением. На «Ямахах» же не было ничего, кроме «защитного» в ПЗУ Бейсика. Я был вынужден умерить свой пыл. Ну а сейчас, когда программно-го обеспечения для «Ямах» становится больше, можно подумать и о расширении курса на будущий год.

Педагогический эксперимент должен был разрешить следующие проблемы: проверка целесообразности достаточно длительного теоретического (безмашинного) вводного курса; апробация содержания темы «Программирование на Бейсике» (графика); зависимость эффективности преподавания от аппаратного и программного обеспечения КВТ. Однако тотальность эксперимента ставила и множество других задач. Практически каждый урок, каждая подтема на уроке являлись экспериментальными. Такое положение дел, с одной стороны, позволило разрешить целый ряд проблем кроме упомянутых, а с другой — оставило много вопросов нерешенными.

Вынужденный экспериментальный анализ первой запланированной проблемы занял все первое полугодие, в котором можно выделить четыре главные темы.

1. Введение. Модель ЭВМ (1—3-й уроки).

2. Системы счисления (3—5-й уроки).

3. Алгоритмы. Блок-схемы алгоритмов (6—10-й уроки).

4. Программирование. Бейсик (11—17-й уроки).

Из них безусловно необходимой можно считать лишь первую. Действительно, учащиеся следуют уже на первых занятиях ввести в курс дела: рассказать о том, что такое информатика, каковы сферы применения ЭВМ и т. д.

Очень важно уже на первых порах ввести понятие алгоритма и объяснить простую модель ЭВМ как универсального исполнителя алгоритмов. Важным методологическим приемом, применявшимся при преподавании этой темы, являются жизненные аналогии. Такое абстрактное понятие, как алгоритм, становится более понятным, если его объяснять на примере похода в магазин; назначение основных узлов ЭВМ — если показать сходство клавиатуры ЭВМ с клавиатурой пишущей машинки, дискового драйвера — с обыкновенным проигрывателем, дисплея — с телевизором. Принцип функционирования ЭВМ можно описать, сравнивая деятельность человека и машины при решении такой задачи, как построение графика функции. В целом, вводя учащихся в новый для них мир информатики, следует использовать максимальное количество знакомых им образов и понятий.

Остановимся подробнее на последнем примере (построение графика функций машиной и человеком). Определив ЭВМ как электронное устройство для ввода, преобразования, длительного хранения и наглядного представления данных, я проиллюстрировал работу всех этих четырех основных функций ЭВМ на примере построения графика некоего физического процесса. Пусть ЭВМ через датчики измеряет его параметры. В этом случае датчики (или, точнее, интерфейсы с ними) будут исполнять функцию ввода данных. Процессор, получив эти данные, может их определенным образом обработать (например, масштабировать) — это функция преобразования данных, сохранить в числовой форме на гибком магнитном диске — это функция долговременного хранения. И, наконец, на экране дисплея или на принтере данные можно представить в графическом виде — функция наглядного представления.

Если теперь график того же процесса строит человек, то легко видеть, что он выполняет те же четыре функции. Я спросил у ребят: в чем они видят разницу между действиями человека и машины? Первые ответы были очевидны: машина делает это быстрее, с большей точностью, аккуратнее. «А есть

ли у человека какие-нибудь преимущества перед ЭВМ?» — продолжал спрашивать я. И вот наконец кто-то отвечает: человек в процессе построения графика физического процесса получает знания об этом процессе, а машина нет. Человек некоторым, логически не вытекающим из полученных данных образом поймет суть происходящего процесса, сможет сделать выводы из наблюдений. Таким образом, сами ребята сделали вывод: машина способна лишь преобразовывать данные, человек же в процессе их переработки получает знания — нечто хотя и зависимое от данных (знаний без данных не бывает), но качественно отличное от них.

Вторая подтема — «Системы счисления» — несла тройную нагрузку. Это, во-первых, знакомство учащихся с двоичной системой счисления; во-вторых, привитие навыков алгоритмической работы на примере перехода из одной системы счисления в другую; в-третьих, демонстрация разницы на примере существования различных систем счисления между знанием (само число, не зависящее от формы записи) и данными (форма записи числа в различных системах счисления). Самостоятельная работа, проведенная на пятом уроке, показала, что в основной своей массе учащиеся усваивают эту тему хорошо.

Надо честно признаться, что темы «Алгоритмы. Блок-схемы алгоритмов» и «Программирование. Бейсик» были усвоены учащимися плохо, за исключением 7—8 человек (в основном тех, кто занимался в кружках программирования или в УПК по специальностям, связанным с ЭВМ). Правда, я ставил в основном хорошие отметки, так как плохими боялся отпугнуть от дальнейшей работы с ЭВМ. Учащиеся не могли написать даже простейших программ и алгоритмов с ветвлением и циклом. Показателен такой пример. В ходе подготовки к занятиям была придумана «задача о толстяках». Смысл ее в том, чтобы с помощью операций сравнения выбрать самого тяжелого человека планеты. В этом занимательном виде задачу решили все, но, как только она была переформулирована (найти максимальный элемент массива), почти никто с ней не

справился. На первый взгляд это странно, однако здесь есть одно различие, которое оказывается решающим. Вот пример. На уроках математики школьники должны решить квадратное уравнение  $x^2 + x = 100$ . Его корни — два совершенно конкретных числа. А на уроках информатики учащиеся получают задание: написать алгоритм решения квадратного уравнения, т. е. записать на определенном языке (блок-схем или алгоритмическом) ту процедуру, которой они пользуются при решении квадратных уравнений. Конкретный ответ здесь тоже есть — это сам алгоритм. Но он для учеников как раз абсолютно не конкретен. Его правильность нельзя проверить ни теоретически (ведь нельзя же давать ребятам на уроках сложную математическую теорию проверки правильности алгоритмов), ни эмпирически — «прокручивать» каждую написанную блок-схему много раз невозможно. Тот факт, что полученный алгоритм с точностью до языка записи повторяет правило решения квадратных уравнений из учебника алгебры, ребят ни в чем не убеждает. С тем правилом понятно, что делать, — используя его, самому решать квадратные уравнения. А с этим... Это мы с вами знаем, что, переводя алгоритм в программу для ЭВМ, мы получим возможность не решать самим уравнения, а передоверить это дело машине. Но школьники этого не знали — ведь машин-то они тогда еще не видели, а поверить мне на слово не хотели, да и не могли. Не случайно все занятия по составлению алгоритмов и программ они называли «теорией», подчеркивая тем самым бессмысленность таких занятий без «практики» — работы на ЭВМ. Только ЭВМ могла конкретизировать эту «теорию».

Кстати, вывод о «неконкретности алгоритмов» подсказали мне сами ребята. В самом начале преподавания темы «Программирование. Бейсик» я задал им два вопроса: «За что вы не любите (любите) информатику?» и «Чтобы вы хотели от информатики в дальнейшем?». Ответы были очень интересными. Один из наиболее типичных и, увы, анонимный: «...Мы делаем какие-то странные вещи, которые трудно как-то применять



в практических целях. Мы усложняем свою жизнь и решение элементарных задач составлением алгоритмов, блок-схем, которые порой бывают очень трудны и громоздки. Я понимаю, что это очень нужно в наше время... Но я еще ни разу не видел применения этих навыков». Вот еще ряд высказываний на эту же тему, на мой взгляд, они очень хорошо проясняют ситуацию. «...Я не понимаю многое в алгоритмах, может быть, потому, что это очень новый тип задач, для которых нужен особый тип мышления» (стиль ответов сохранен. — Д. С.). Из ответов на второй вопрос: «Чтобы побольше работать на ЭВМ. Чтобы объясняли понятнее, с более конкретными примерами («Уж куда конкретнее!» — думал я тогда. — Д. С.), которые можно записать на ЭВМ почему и как. При работе с ЭВМ материал лучше и легче запомнится и понятнее». «Хотели бы, чтобы тема «Бейсик» объяснялась более понятно и чтобы нам рассказывали, а лучше показывали, какая связь всего того, что мы изучаем, с машиной. Вообще пока что эту новую тему на уроке понять довольно трудно. А сама при составлении программы я просто не понимаю, как проверить правильность ее».

Подытоживая, можно сказать, что перед началом машинного курса следует давать учащимся теоретическое введение в размере 1—2 уроков — не более. Темы «Алгоритмы. Блок-схемы алгоритмов» и «Программирование» следует проходить только с использованием ЭВМ, сочетая теоретическую и практическую работу.

Во втором полугодии началась работа в КВТ. Сразу же выяснилось, что провести эксперимент по третьему пункту плана в полном объеме не удастся, так как необходимое программное обеспечение почти полностью отсутствовало; единственно возможной темой изучения был язык программирования Бейсик. Зато возникла другая проблема: в каком порядке следует знакомить учащихся с ЭВМ и основными конструкциями языка Бейсик. Была выбрана следующая последовательность: знакомство с клавиатурой (2 урока); отладка уже написанных в предыдущем полугодии программ (4 урока); знакомство с графикой (8

уроков); итоговая контрольная работа (2 урока). И вот что я узнал в итоге.

Нецелесообразно отводить на знакомство с клавиатурой 2 часа — это очень много. Учителю стоит примерно в течение 10 минут рассказать об основных клавишах и дать 15—20 минут на тренировку (желательно с помощью какого-либо клавиатурного тренажера). Дальнейшее освоение клавиатуры учащимися будет происходить постепенно в процессе работы по отладке программ. Это хороший способ — в будущем школьники, которые столкнутся с новыми типами клавиатур, будут осваивать их именно таким образом.

Неэффективность безмашинного обучения программированию была наглядно доказана на занятиях, посвященных «отладке» ранее написанных программ. Учащиеся продемонстрировали лишь большую степень «бейсикобоязни» и «бейсиконепонимания» (разумеется, здесь речь не идет о тех, кто имел предварительный опыт работы на ЭВМ). В целом эти уроки еще раз доказали недопустимость разрыва между теорией и практикой при преподавании программирования.

А вот знакомство с графикой дало взрыв интереса к урокам информатики. Оно происходило в следующем порядке. Вначале школьники познакомились с четырьмя операторами графики (SCREEN, LINE, PSET, CIRCLE) в самой простой форме (т. е. без параметров типа цветность, масштаб и т. д.). На это ушло примерно полчаса. Все остальное время сдвоенного урока посвящалось практической работе. Задание было такое: с помощью только что объясненных операторов учащимся предлагалось построить что-нибудь, что походило бы на картинку из учебника по геометрии. Энтузиазму не было предела. Получив наконец-то возможность сделать что-нибудь самостоятельно и, главное, наглядно-конкретное, ребята за час нарисовали множество интересных чертежей, причем не ощущалось никакой разницы между «отличниками» и «двоечниками». И те, и другие с увлечением работали, добиваясь очень неплохих результатов. Правда, у «отличников» (т. е. дисциплинированных и аккуратных учени-

ков) интерес, как правило, был устойчивым, а у «двоечников» он (тоже, как правило, хотя были и исключения) скоро прошел.

После этого ошеломляющего успеха у меня возникла идея «контрабандой» дать все ранее не очень хорошо понятые операторы Бейсика (прежде всего FOR...NEXT и IF), и на следующем занятии вместо того, чтобы расширять познания учащихся собственно в области графики (например, объяснить параметры B и BF в операторе LINE, рассказать об операторе DRAW и т. д.), я объяснил, как с помощью операторов FOR...NEXT и CIRCLE закрашивать окружности. Возникающий после этого целый спектр задач (закрашивание прямоугольника, рисование «бублика») был успешно решен уже самими школьниками в ходе практической работы.

Следующий двойной урок, построенный по схеме двух предыдущих, был посвящен движению объектов на экране. Только теперь учащиеся получили сведения о возможности использования цветной графики — необходимость закрашивания предыдущего изображения цветом фона диктовала это — и пересылки программ с машин учеников на машину учителя — желание посмотреть свое произведение в цвете было хорошим стимулом для быстрого запоминания команд, связанных с пересылкой по сети. Конкретная задача — изменение направления движения точки при встрече с преградой — позволила еще раз объяснить действие оператора IF.

Следующее занятие и контрольная работа были в основном посвящены закреплению пройденного. На контрольной нужно было нарисовать прохождение луча света в двух различных средах. В целом учащиеся успешно справились с этой задачей.

Итак, тема «Графика» была усвоена лучше всего. Более того, можно сказать, что настоящее знакомство учащихся с информатикой началось именно с нее.

Поэтому преподавателям информатики можно посоветовать начинать обучение языку Бейсик с графики; именно на примере рисунков следует давать такие операторы, как FOR...NEXT и IF. И лишь после твердого усвоения учащимися этой темы можно переходить к решению различных числовых задач. (К сожалению, эти задачи существенно менее интересны, чем связанные только с графикой, поэтому их следует давать в как можно более занятой форме. Придумывание неординарных постановок обычных задач является, на мой взгляд, самой важной задачей преподавателей Бейсика. При этом можно воспользоваться богатым опытом составления подобных задач в элементарной математике.)

Закончить статью мне бы хотелось высказыванием одного из моих подопечных. «В принципе, информатика предмет интересный. Но ее ввели совсем недавно, а годами складывались совершенно другие «серьезные» предметы. Поэтому, думаю, информатика, конечно, не сложившийся предмет. Все (или почти все) приходят на нее отдохнуть, заняться списыванием или еще чем-нибудь. Если бы все слушали ваши объяснения с вниманием (с начала года), то, думаю, информатика просто понравилась бы всем, а с вводом ЭВМ тем более. Ее не любить просто нельзя. Но, повторяю, трудно в себе перебороть тот настрой, с каким ты на нее приходишь. Во многом это зависит от учителя. Но ничего плохого о вас, Дмитрий Александрович, я не думал. Вы учитель начинающий, неопытный, и это придет со временем. Сама информатика лет через 5—6, думаю, станет привычным для всех предметом, это дело времени. А пока...» Друзья, давайте докажем школьникам, что для становления нашего предмета требуется меньше 5—6 лет. А педагогический опыт лучше всего приобретать в конкретном деле.



## Проблемы нового курса

В настоящее время в распоряжении преподавателей информатики имеется пробный материал и методическое пособие, рассчитанное на безмашинное обучение. Школьные кабинеты информатики оборудованы по-разному, их можно разделить на три категории по технической обеспеченности:

имеющие на каждого обучаемого автоматизированное ученическое место, обеспеченное микро-ЭВМ или дисплейным терминалом, общим для всего класса ЭВМ;

не имеющие вычислительной техники; преподавание в этом случае базируется на «плакатном» материале;

оснащенные техникой частично, как правило, программируемыми калькуляторами (не менее одного калькулятора на двух учеников) и несколькими микро-ЭВМ.

К первой категории относятся экспериментальные классы, уже несколько лет работающие в опытном порядке, а также кабинеты, созданные на базе УПК, готовые операторов-программистов. У них накоплен опыт преподавания и имеются соответствующие методические материалы.

Вторая категория — наиболее многочисленная. Естественно, что таким школам уделяется повышенное внимание и в плане подготовки учителей, и в плане разработки учебных пособий. Уже сейчас они стараются обеспечить себя калькуляторами, а в дальнейшем будут иметь и микро-ЭВМ. Следовательно, количество школ, не имеющих вычислительной техники, будет сокращаться.

Пока немногочисленна третья категория: Это школы, в которых шефствующие предприятия и организации смогли установить некоторую вычислительную технику. Как правило, техника разнотипна и кабинеты информатики отличаются и по количеству, и по маркам ЭВМ. Из-за своей малочисленности эта категория не пользуется достаточным методическим и организационным вниманием. По нашему мнению, это ошибоч-

ная точка зрения. Количество таких кабинетов информатики действительно пока невелико, однако через эту стадию должны будут пройти почти все школы. Поэтому можно утверждать, что число их будет возрастать и существование такой категории — явление длительное. Необходимо также отметить особую сложность перехода от безмашинного к обучению, обеспеченному техникой частично. Более того, работа в таких кабинетах сложна и для выпускников пединститутов, если они будут ориентированы на преподавание в автоматизированных классах.

В школе № 116 г. Куйбышева кабинет информатики оснащен программируемыми калькуляторами и пятью микро-ЭВМ типа «Электроника ДЗ-28». Преподавание базируется на программе курса «Основы информатики и вычислительной техники», которая со временем будет совершенствоваться и в той или иной степени перерабатываться, но пока является обязательной. Составлена программа довольно плотно. Если ее придерживаться полностью, то времени на дополнительные темы не остается. Ориентированы программа и учебник на безмашинное обучение, поэтому использование любой техники без значительных дополнительных разъяснений невозможно. То есть необходимо изучать соответствующие разделы учебника, решать примеры и записывать в тетради выполнение машины соответствующих вычислительных операций. В результате возникает проблема: как при этом вызывать и постоянно поддерживать у учеников интерес к предмету, если в кабинетах есть, но не используется вычислительная техника?

К работе непосредственно на микро-ЭВМ девятиклассники не готовы. Кроме того, ЭВМ всего пять, работать с ними придется по очереди, то есть тот же безмашинный вариант обучения в ожидании очереди для работы с ЭВМ. Мы пошли по пути, который избрали многие учителя информатики, — использова-

ние на уроках программируемых микро-калькуляторов, которыми обеспечены все ученики. Применение техники дало соответствующий психологический эффект. Первые успехи в решении задач на калькуляторе — не очень сложном в обращении, но достаточно «умном» устройстве — воодушевили большинство учащихся. Поэтому обычные методы поддержания интереса к предмету практически не требуются. Одновременно с помощью сотрудников Куйбышевского института инженеров железнодорожного транспорта решена проблема применения на уроках микро-ЭВМ. Разработаны программы, позволяющие контролировать усвоение отдельных тем курса, а также правильность решения типовых задач. При этом в случае необходимости на дисплее даются разъяснения и разбор вариантов решения. В конце диалога с ЭВМ выставляется оценка, учитывающая количество попыток решения задачи.

Уроки проходят следующим образом: рассматривается очередная тема, решаются примеры и задачи, предложенные в учебнике. Затем разбираются варианты решения такого типа на калькуляторе, для него составляется и обсуждается программа. Проверенная программа вводится в калькулятор, подставляются числовые значения, выполняется решение примеров. При таком методе увеличивается количество времени, необходимое для изучения каждой темы. Как правило, одна тема рассматривается дважды: на алгоритмическом языке и с использованием системы команд калькулятора. Основная сложность заключается в трудоемкости составления программ. То, что в терминах алгоритмического языка выглядит просто и понятно (команды ветвления, повторения, использования табличных величин), на калькуляторе реализуется большим количеством команд. Приходится дополнительно изучать условные и безусловные переходы, организацию подпрограмм и обращений к ним, косвенную адресацию вызова и записи чисел в регистры памяти. Затраты времени такие же, как и на объяснение аналогичных задач в терминах алгоритмического языка.

Для того чтобы сэкономить время, мы отказались от традиционных форм опроса. Разбор домашнего задания ведется только в том случае, если нужно закрепить материал перед изучением следующей темы, или заданные примеры имеют принципиальное значение. Во всех остальных случаях проверка ведется с помощью микро-ЭВМ. В ЭВМ моделируется та же задача, которую решает ученик. Это может быть составление алгоритма — машина проверяет правильность выбора команд из предлагаемого перечня и их размещение в определенной последовательности — или решение какого-либо примера на калькуляторе — числовые значения генерируются ЭВМ с помощью датчика случайных чисел и проверяется полученный ответ. Микро-ЭВМ работает в контрольно-обучающем режиме. При верном выполнении задания ЭВМ выдает следующее. Если же допущена ошибка, то она анализируется, на экране дисплея появляется подсказка и предлагается снова выполнить задание. При повторной ошибке подсказка более развернута. Если же ученик не может справиться с решением, то выдается рекомендация обратиться к преподавателю. За 15—20 минут с помощью пяти микро-ЭВМ опрашивается 10—15 человек. Количество опрошенных колеблется в зависимости от сложности и уровня подготовки учеников. За 2—3 урока, отводимые для изучения одной темы на ЭВМ, проверяют свои знания практически все учащиеся; получившие неудовлетворительные оценки при этом вызываются повторно. Программы опроса разрабатываются самими школьниками членами клуба юных программистов, под руководством работников вычислительного центра, курирующих преподавание информатики в школе. В результате проводится полный контроль усвоения учениками каждой темы. Школьники приобретают навыки работы с микро-ЭВМ, что в дальнейшем поможет освоить принципы программирования и использования любого типа ЭВМ.

По нашему мнению, это наиболее перспективная методика преподавания информатики в школах, оснащенных вычислительной техникой частично. Одна-



ко для широкого использования необходимо решить следующие проблемы: обобщить опыт преподавания информатики с использованием микрокалькуляторов, чтобы определить перечень тем и время, которое потребуется для их изучения;

разработать минимум, который должны усвоить ученики, т. е. утвердить «обязательные результаты обучения», чтобы, варьируя пути изучения информатики, все преподаватели шли к единой цели;

срочно собрать и тиражировать как в печатном виде, так и на машинных носителях контрольно-обучающие программы на микро-ЭВМ и программы для калькуляторов, которые помогут учителю выбрать необходимый вариант в зависимости от набора технических средств, степени усвоения темы и других условий. На наш взгляд, уже назрела необходимость создания отраслевого фонда алгоритмов и программ Министерства просвещения по аналогу других отраслей народного хозяйства.

Несколько слов о программе курса «Основы информатики и вычислительной техники». Разумеется, она является пробной и будет совершенствоваться. Однако уже сейчас неразумно уделять в X классе столько времени изучению устройства ЭВМ. Вполне достаточно общих сведений о двоичной системе счисления и объяснения принципов выполнения в машине арифметических и логических операций. В то же время изучение структуры ЭВМ (особенно организации запоминающих устройств) желательно перенести в про-

грамму IX класса. Возможно, действующий вариант программы наиболее подходит для безмашинного обучения. Однако использование на уроках реальных ЭВМ, начиная с микрокалькулятора, невозможно без уяснения принципов работы ЭВМ.

И еще об одной проблеме. Обучая школьников информатике, мы готовим в первую очередь будущих пользователей ЭВМ, которые будут работать во всех сферах. Однако примеры, показывающие возможности ЭВМ, взяты в основном из математики, физики, химии. Это расчетные задачи. Не умаляя значения такого типа задач, хотелось бы указать на опасность представления информатики как некоего приложения к перечисленным дисциплинам. В то же время в программу не включены сведения о большом классе автоматизированных информационных систем — поисковых, справочных, экспертных и др. Но ведь именно с такими системами будут чаще сталкиваться сегодняшние ученики. Объяснить принципы работы таких систем сложно, а организовать достаточно работоспособный банк данных на используемых сегодня микро-ЭВМ крайне трудно. И все-таки следует подумать о разработке моделей, позволяющих имитировать работу информационных систем различной тематики. Тогда значительно проще решился бы вопрос об использовании вычислительных машин как средства обучения и в программу нового курса можно было бы ввести такие интересные темы, как решение задач из курса биологии, литературы, истории, иностранного языка.



*«Здравый смысл — это врожденная способность интуитивно отличать правду от лжи и видеть вещи в их истинном свете».*

Французский энциклопедический словарь Ларусса

В мае 1990 г. старший инженер А. Б. Сидоров, работник отдела снабжения автоВАЗа, собирался в командировку. В его задание входили переговоры с двумя заводами — Свердловским и Челябинским, которые поставляли комплектующие изделия к новым моделям вазовских автомобилей.

Начальник отдела поручил снабженцу перед командировкой проверить, нет ли в этих городах других поставщиков ВАЗа, и заодно заехать туда для решения злободневных вопросов. В отделе снабжения стоял ПК, соединенный линией связи с главным информационным центром ВАЗа. Сидоров решил запросить нужные сведения у ЭВМ, как он это неоднократно делал раньше. Только вчера он набирал на дисплее вопрос: «Откуда поступают фары для модели ВАЗ-2109?» — и ПК почти мгновенно высветил на экране наименование предприятия, адрес и телефон отдела поставок.

Вот и сегодня он набрал запрос о поставщиках, расположенных в городах, которые ему предстоит посетить, но у ЭВМ сегодня было «плохое настроение» — ответы были отрицательными, сколько он ни повторял вопрос. Позвонив по телефону в вычислительный центр, раздраженный снабженец потребовал заменить «сломавшийся» компью-

# ИЛИ ЧТО ТАКОЕ ЗДРАВЫЙ СМЫСЛ

А. ХАЛАМАЙЗЕР,  
М. ПЕРЕГУДОВ

тер. Он объяснил, что неоднократно получал на свой запрос бессмысленные ответы. По жалобе прибыл консультант-программист и предложил снабженцу еще раз набрать на клавиатуре его запрос. Тот набрал «поставщики, расположенные в Свердловске и Челябинске». На экране сразу же появилась надпись: «не значатся».

— Вот видите?! — кипятился Сидоров. — А я сам точно знаю, что и в том, и в другом городе у нашей фирмы есть поставщики!

Консультант попытался было объяснить незадачливому снабженцу, что ни один завод не может быть расположен одновременно и в Свердловске, и в Челябинске.

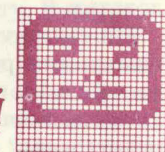
— Вы должны сформулировать, что именно вы хотите получить в ответ на ваш запрос, — терпеливо объяснял консультант.

— Перестаньте учить меня, что я хочу получить! — вновь обиделся снабженец. — Это я сам знаю гораздо лучше вас! Я хочу, чтобы в списке был каждый поставщик из Свердловска и каждый поставщик из Челябинска!

— Сейчас вы получите этот список, — ответил консультант.

Он набрал «...В Свердловске или Челябинске», и секунду спустя на экране появились строчки с адресами; затем он

Веселый  
урок







нажал клавишу «печать», и через полминуты вручил жалобщику требуемый список.

124 — А если вы хотите получить отдельные списки поставщиков из Свердловска и из Челябинска, — продолжал консультант, — то лучше набрать два отдельных запроса.

Минуту спустя компьютер напечатал два отдельных списка.

Люди думают и говорят на человеческом языке (английском, немецком, русском...), а не на языке формальной логики. Человеческий язык несравненно богаче формального языка, он насыщен различными оттенками, но обращаться к компьютеру на этом языке пока еще невозможно: он недостаточно точен. Компьютер, в отличие от человека, не наделен здравым смыслом. Он не может догадаться, что требуется список всех поставщиков, расположенных в Свердловске, и всех, расположенных в Челябинске, но не тех, что расположены и в Свердловске, и в Челябинске, т. е. сразу в двух местах.

### Компьютер, к сожалению, слишком точен

Говорят, что на прошении одного осужденного о помиловании самодержец всероссийский собственноручно начертать соизволил: «казнить нельзя помиловать».

Но скажите, читатель, как следует понимать эту резолюцию: казнить, потому что нельзя помиловать, или, наоборот, казнить нельзя и осужденного следует помиловать?

«Человеческий» язык оказывается порой не вполне ясным и у великих. Помните «Песнь о вещем Олеге»?

*Как черная лента вокруг ног обвилась,  
И вскрикнул внезапно ужаленный князь.*

Скажите, читатель, к чему здесь относится «внезапно»? Внезапно, т. е. неожиданно для окружающих, вскрикнул? Или, может быть, внезапно ужаленный? Разбирая предложение, нельзя получить ясного ответа на этот вопрос. Правда, интонация пушкинского стихотворения позволяет думать, что поэт имел в виду второй вариант (внезапно ужаленный). Но какое дело компьютеру до интонации?

— Марина, запиши на доске, — диктует учительница, — пять плюс два умножить на три. Записала? Сколько получилось?

— Получилось одиннадцать, Ольга Петровна.

— Неверно! — вскакивает Сережа с соседней парты. — У меня получилось двадцать один!

Оба ученика правы! Различия в ответах связаны с тем, что учительница не сказала, нужно ли умножать на три всю сумму (пять плюс два), или нужно умножить только двойку, а потом к пяти прибавить произведение (два на три).

А вот «трехсмысленная» фраза: «Витя встретил Нину на поляне с цветами». Догадайтесь, читатель, к чему здесь относится «с цветами».

Если такая фраза встретится в художественном произведении, читатель начнет догадываться, какой из трех смыслов имел в виду автор. Не догадавшись, отметит про себя небрежность автора — и только. Но при описании алгоритма, при составлении программы для ЭВМ такие двусмысленности (или даже «трехсмысленности») недопустимы: машина выполнит совсем не то, что имел в виду программист.

Одним из следствий компьютеризации является то, что аккуратность в формулировании правил (законов, постановлений, порядка выполнения тех или иных действий) приобретает огромное значение. При отсутствии ЭВМ плохо сформулированные правила быстро «корректируются» исполнителями — их просто перестают выполнять буквально, а дела-

ют то, что считается нужным на месте. Те же самые правила, заложенные в компьютер, становятся источником многих неприятностей: компьютер ведь не наделен пока «здравым смыслом».

### Счет на 00.00 рублей

Представим себе сотрудника «Мосэнерго», который следит за оплатой по электросчетчикам, руководствуясь следующим правилом:

*«По каждому клиенту ежемесячно вычесть старые показания счетчика из новых, умножить результат на цену kilowatt-часа, выписать счет. После оплаты вычеркнуть копию счета из списка. Если счет не оплачен, выписать напоминание, а если он не оплачен после этого, отключить подачу энергии.»*

Все будет в порядке. Но однажды обязанности сотрудника, выписывавшего счета, перепоручили компьютеру, для которого именно по этому правилу составили программу. И для первого клиента, который в течение всего месяца отсутствовал, компьютер выписал «счет» на 00.00 рублей, через месяц выслал напоминание, а через два — отключил подачу тока. Говорят, что пришлось срочно оплатить счет на вышеуказанную сумму, а затем исправить ошибку в программе. Одновременно была внесена поправка и в правило: «Если за месяц энергии потреблено меньше, чем на 00.01, то счет не выписывается».



Совершенно очевидно, что программист, забыв о «здравом смысле», чересчур буквально заложил в программу сформулированное выше правило.

### ЭВМ поправляет бухгалтера

Месячный оклад инженера Иванова составляет 200 рублей; непрерывный стаж его работы — больше 8 лет, так что по больничному листу он получает все 100 %. Пока зарплату считали вручную, ему всегда, независимо от того, болел ли он в данном месяце, выписывали ровно 200 рублей.

Однако после внедрения ЭВМ, когда Иванов проболел 3 дня из 22 рабочих дней апреля 1985 года, компьютер «решил» иначе.

среднедневной заработок. . . 200.00:22=9,09  
за 19 проработанных дней 9,09×19=172,71  
за 3 дня болезни . . . . . 9,09×3=27,27  
всего начислено за месяц . . . . . 199,98

Получив напечатанный компьютером расчетный листок, инженер посмеялся, но требовать 2 копейки не стал. В бухгалтерии пытались было обвинить авторов программы в ошибке, но оказалось, что правила соблюдены программой точно, полагается считать именно так. А вот бухгалтерия вручную уже 20 лет считала неправильно, хотя проболевшему часть месяца работнику исправно начисляется его должностной оклад копейка в копейку.

В следующем месяце программа автоматически удержала с Иванова не 21.20 подоходного налога, как обычно, а 21.07, так как правило требует облагать налогом только полные рубли заработной платы! Тут уже бухгалтерия всполошилась: одно дело — недоплатить две копейки, тем более что в суд инженер не подал, а другое дело — переплатить 13 копеек! Но оказалось, что и здесь программа действует по букве закона. Решение ЭВМ было оставлено в силе.

В обоих описанных случаях ЭВМ принимала не такие решения, как люди, ранее выполнявшие ту же работу. В первом случае оказалось, что не прав разработчик программы, недостаточно изучивший правила, которые не были четко сформулированы, а существовали только в виде навыков сотрудников. А во втором



случае неправыми оказались люди, выполнявшие правила до внедрения ЭВМ неточно. Рассмотрим теперь третий случай, когда ЭВМ поставила людей перед необходимостью изменения правил, так как оказалось, что эти правила по формальным мотивам невыполнимы.

В министерстве выдавали зарплату за прошедший месяц первого числа. Для этого нужно было уже утром 30 числа предыдущего месяца иметь готовые расчеты и итоговые суммы, чтобы заказать деньги в банке. Но для получения этих результатов от ЭВМ нужно хотя бы накануне иметь для ввода в нее все данные по табелю. Иными словами, иметь 29 числа сведения о том, кто отработал (и полный ли день) 30 числа! До внедрения ЭВМ это делалось неформально и за точность никто не отвечал, так как уже потом можно было «оперативно» вручную подправить сумму, выдаваемую заболевшему или прогулявшему 30 число

работнику. Для ЭВМ же данные нужно было готовить в виде строгих документов определенной формы с подписью табельщика. Табельщики отказывались раньше, чем к середине дня 30 числа, подписываться под данными о выходе на работу сотрудников за целый месяц. Они справедливо утверждали, что нельзя подавать в ВЦ сведения о работе за завтра. А если подать эти сведения завтра, то деньги не будут выплачены к 1 числу, так как их просто не успеют заказать и получить в банке.

Администрацией было принято единственное возможное решение: выплачивать зарплату не 1, а 3 или 4 числа. До внедрения ЭВМ бухгалтерия начисляла зарплату всей тысяче сотрудников в конце месяца «авансо», так как, строго говоря, было неизвестно, доработает ли человек до конца месяца, а ему уже начислили оклад. Это, конечно, было нарушением; правило выплаты зарплаты 1 числа было невыполнимым, и внедрение ЭВМ явилось причиной его пересмотра.

## Напечатано в 1986 г.

Встреча с читателями № 3  
Совещание в ЦК КПСС № 2  
Щербаков С. Г. Дело государственной важности № 1

### В МИНИСТЕРСТВЕ ПРОСВЕЩЕНИЯ СССР

Кабинет вычислительной техники всех типов средних учебных заведений (на базе персональных микро-ЭВМ). Методические рекомендации № 3  
О доплате за заведование кабинетом вычислительной техники и за работающие компьютеры № 2  
О подготовке руководящих и педагогических кадров народного образования по основам информатики и вычислительной техники № 2  
Об изучении материалов XXVII съезда в учебных предметах общеобразовательной школы № 1

### ОБЩИЕ ВОПРОСЫ

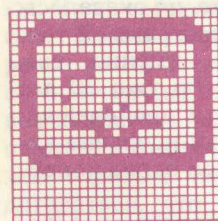
Баринев А. Компьютер в ПТУ: сегодня и завтра № 3  
Велихов Е. Новая информационная технология в школе № 1  
Волков Д., Ширков П., Деянов Р. Опыт ускоренного обучения № 3  
Кобринский Я., Кузнецов А. Особенности пакетов прикладных программ № 3  
Ковалев А., Хорошева И. Итоги, проблемы, перспективы № 1  
Конференция в Москве № 1  
Лисов В., Парамонов А. Комсомольцы и компьютерный всеобуч № 2  
Мельников В. С. А. Лебедев — основоположник отечественной вычислительной техники № 1  
Орешков И. Информатика в техникуме № 1  
Смирнов С. Нужен ли компьютер на обычном уроке? № 2  
Труш В., Голод В. Компьютеры в дефектологии № 2  
Уваров А. ЭВМ на пути в школу № 1  
Федоренко Н. Математические методы управления народным хозяйством № 1  
Филимонова Л. Учить по-новому № 2  
Хорошилов В., Красс Э. Профессиональное образование и компьютерный всеобуч № 1  
Шипунов В. Работая сообща № 1  
Школа в Пушкино № 2

### МЕТОДИКА ОБУЧЕНИЯ

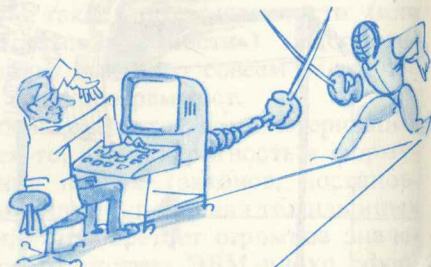
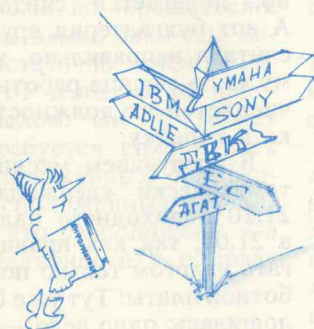
Ермакова В., Утлинский Е. Лабораторный практикум «Алгоритмический язык» № 2  
Завич Л., Самовольнова Л. От экрана телевизора — к экрану дисплея № 1  
Методические указания по преподаванию курса «Основы информатики и вычислительной техники» в X классе № 1—3  
Положение об условиях проведения конкурса по созданию учебника «Основы информатики и вычислительной техники» для средних учебных заведений № 1  
Программа курса «Основы информатики и вычислительной техники» (для объявления конкурса на создание учебника) № 1  
Хорошева И., Гольц Я., Шень А. Рекомендации по преподаванию курса «Основы информатики и вычислительной техники» № 1

### КАБИНЕТ ВТ

Абрамов И. Имитатор микропроцессора № 2  
Боканс Я., Грибустс А. На базе БК-0010 № 2  
Бояринов А. Машинная поддержка курса № 2  
Варсановьев Д. Альфа-практикум № 2  
Варсановьев Д., Дымченко А. Е-86 № 1  
Викентьев Л., Козлов О., Костров А. ЕС ЭВМ в курсе информатики № 3  
Виллемс А., Пейал Я. Советы по созданию дружественного программного обеспечения № 1  
Гельтишцева Е. Гигиена труда в КВТ № 1  
Денисенко А. Испытания КУВТ-86 № 2  
Дрига И. Оборудование для КВТ № 2  
Дуванов А. Какой язык лучше? № 2  
Ершов А. Решение задач с применением программируемого микрокалькулятора «Электроника БЗ-34» № 3  
Иканин Е. Основы языка Бейсик № 3  
Козбенко О. Первые итоги обнадёживают № 1  
Кузнецов А., Сергеева Т. Обучающие программы и дидактика № 2  
Кушниренко А., Семенов А. Программы для школьного компьютера № 1  
Леонova Л., Саватеева С. Занятия в КВТ: влияние на организм № 3



Веселый урок



Рисунки Ю. и С. Весковых, И. Печенева



Огурцов А. Игра, обучающая работе с клавиатурой № 2

Перечни технических средств, учебно-наглядных пособий и мебели для кабинетов вычислительной техники № 1

Роберт И. Какой должна быть обучающая программа? № 2

Соловьев С., Дубовой В. Освещение дисплейных классов № 1

Сулим М., Лазарев А., Сафонов В. Знакомьтесь: «Корвет» № 1

Федоров Л., Шадрин Г. Обучающая программа «Машина Поста» № 1

Шилов В. Широкие возможности микрокалькулятора № 2

## ВНЕКЛАССНАЯ РАБОТА

Антипов И. Управляемый карандаш № 3

Григас Г. Программирование — заочно № 2

Добин В. Технику создают люди № 2

Пачиков С. Дома за дисплеем № 3

Туманова О., Лихачева Л. «Иначе бы мы к вам не пришли...» № 3

Флеров М. «Черепашка» учит программировать № 3

## ЭВМ В НАРОДНОМ ХОЗЯЙСТВЕ

Поляков В. На суше и на море № 3

Стрепихеев В. Компьютеры для агропрома № 1

Таланов В. Компьютер приходит в цех № 2

## РЕПОРТАЖ НОМЕРА, ИНФОРМАЦИЯ

№ 1—3

## ЗАРУБЕЖНЫЙ ОПЫТ

Афанасьев Г. «Галактика» — школьникам № 2

Лобовкина М. У наших друзей № 1

Сендов Бл. Инструмент обучения — компьютер № 1

## ПЕДАГОГИЧЕСКИЕ КАДРЫ

Маргынченко О., Колосова Т., Берестичевский С. Учитель и компьютер № 1

Ратинский М. Курсы для учителей № 3

## СЛОВАРЬ ИНФОРМАТИКИ

Словник учебного терминологического словаря «Основы информатики и вычислительной техники в средних учебных заведениях» № 1

Шанский Н., Окунева А., Баско Н. Терминология курса № 1

## ВЕСТИ С МЕСТ

Авербух В., Сенокосов А. Олимпиады по программированию № 1

Далингер В. Что показал опрос учителей № 2

Попов В. Новый курс в нашей школе № 2

Ускова О., Горбенко О. Педагоги учатся № 2

Черникова И. Для уроков информатики № 1

## НАМ ПИШУТ

Славин Д. Каждый урок — эксперимент № 3

Тутаринов Ю., Чиняев А. «Пропагандист» № 3

Юшков С., Андрофагин А. Проблемы нового курса № 3

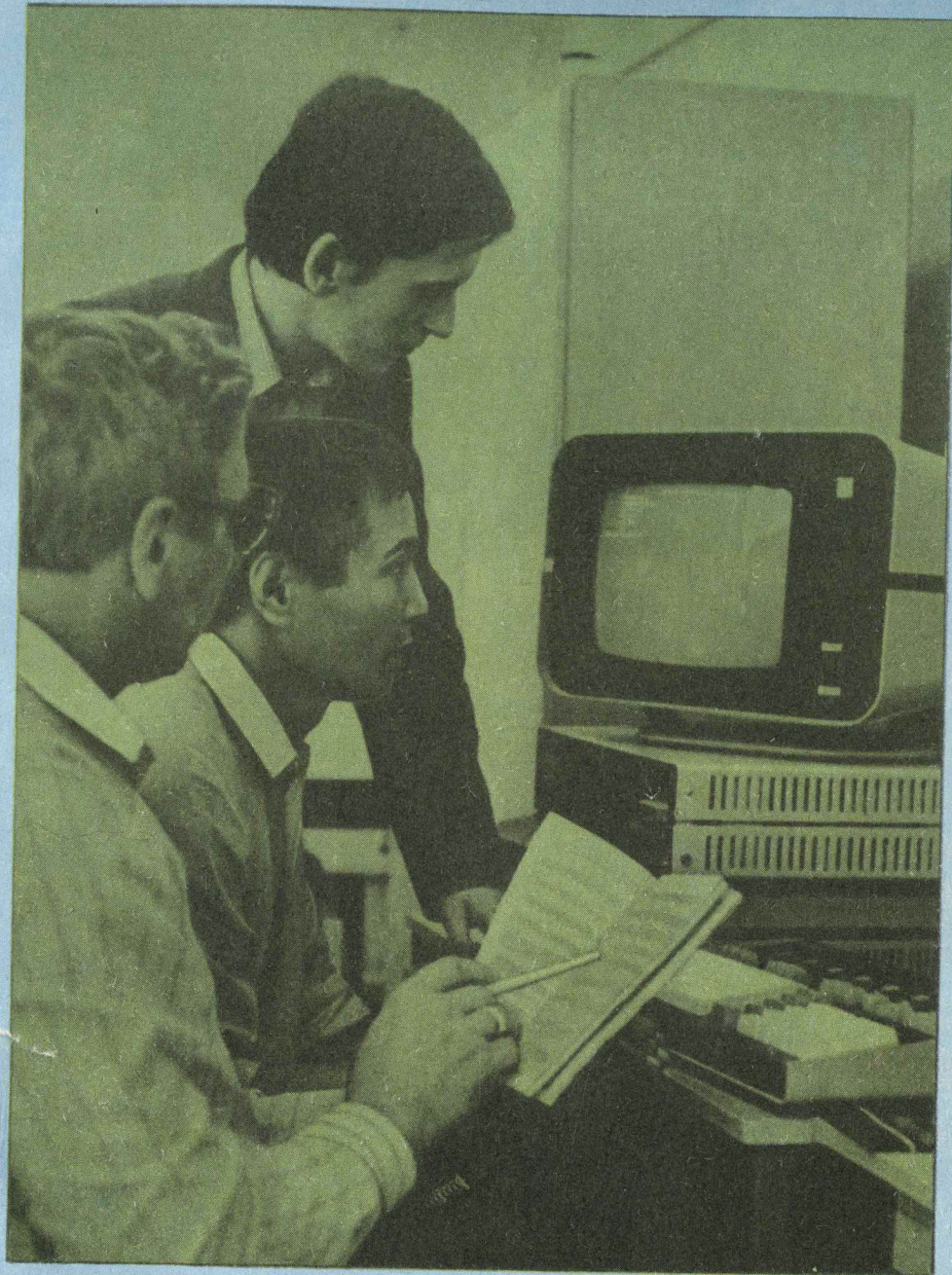
## БИБЛИОГРАФИЯ

Книги, которые будут выпущены в 1987 г. в издательстве «Педагогика» № 1

Машбиц Е. Компьютеризация обучения: проблемы и перспективы № 1

## ВЕСЕЛЫЙ УРОК

№ 1—3





# ИНФ

## 3 '86

### ИНФОРМАТИКА И ОБРАЗОВАНИЕ

