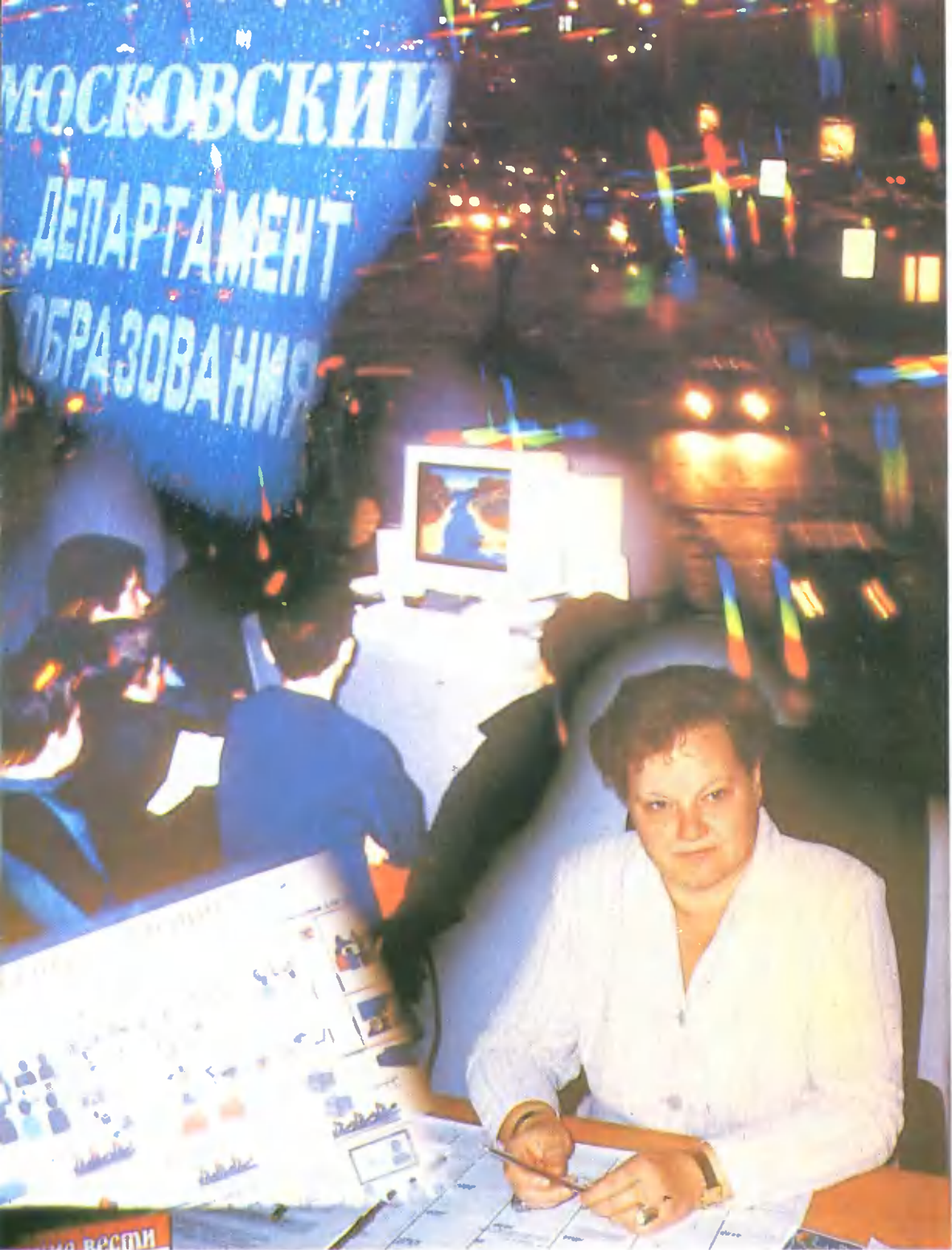


ИНФ  
5 1995

ИНФОРМАТИКА  
И ОБРАЗОВАНИЕ





**Первый заместитель Председателя  
Московского департамента образования  
ЛАРИСА ЕВГЕНЬЕВНА КУРНЕШОВА**



# МОСКОВСКИЙ ВЫПУСК

## Л.Е.КУРНЕСОВА: ОТ ИНВЕСТИЦИЙ В ОБРАЗОВАНИЕ ЗАВИСИТ БУДУЩЕЕ РОССИИ

Москва готовится к 850-летию. Это очень важная дата не только для нашего города, но и для всей России, потому что Москва — столица Российского государства, а это обязывает. В трудные для России годы именно Москва выступала с решающими инициативами и брала на себя ответственность за судьбу государства.

Я думаю, что образование столичного региона тоже имеет свою значимость и играет определенную роль для других территорий Российской Федерации.

Правительством Москвы утверждена программа «Столичное образование», в которой определены соответствующие цели. Одна из них — внедрение современных наукоемких технологий непосредственно в сферу образования, другая — изучение этих технологий в образовательных учреждениях.

Оба направления являются необходимыми элементами для создания стабильного будущего в России.

Поэтому основная стратегия Московского департамента образования — переход на новые образовательные технологии — ставит перед нами очень серьезные задачи по отношению к информатизации, требуя пересмотра подходов к процессу оснащения школ компьютерной техникой и другими видами технических средств.

В начале компьютеризации, а для Москвы этот процесс начался примерно двадцать лет назад, мы должны были быстро оснастить школы любыми компьютерами и обучить детей работать на них. Именно так формулировалась задача со стороны вышестоящих инстанций и партийных органов.

Мне кажется, что в то время это было разумно.

За этот период в Московском регионе накопился значительный и интересный опыт по использованию информационных технологий в образовании, и в настоящее время мы рассматриваем компьютер как элемент новых образовательных технологий, предъявляя к нему, соответственно, новые требования.

Компьютер в современной школе — это не только компьютерный класс, не только место, где ребенок может решать простейшие задачи, набирать и редактировать текст и т. д., а, в том числе, и средство, с помощью которого мы можем создать современные в технологическом отношении информационные среды для управленцев всех уровней, для учителей и для ученика — основного участника учебного процесса.

При таком подходе необходимы нетрадиционные решения относительно конфигурации при размещении компьютеров в информационной среде. Нужны нетрадиционные разработки и методики по использованию компьютеров в предметах общеобразовательного цикла, в кружковой работе, в проектных формах организации учебной деятельности, которые очень перспективны для многих педагогических коллективов. Встают вопросы об эффективной организации информационных сред и их взаимодействиях. Особое место компьютер должен занять в школьной библиотеке, так как при этом значительно расширяются возможности доступа к информационному фонду для учителей и учащихся. Есть свои проблемы и при использовании компьютера в

сфере дополнительных образовательных услуг.

Таким образом, мы считаем, что компьютерный класс в школе на современном этапе развития информатизации образования как один из элементов этой информатизации может существовать, но как совокупность, обеспечивающая всю образовательную среду, он, безусловно, недостаточен.

Учитывая, что все эти процессы не простые и не имеют однозначного ясного решения, Московский департамент образования создает сеть экспериментальных площадок, на которых будут отработываться целостные модели, в частности, информационного пространства управленческого образовательного процесса учебного заведения.

Таким образом, можно констатировать, что сегодня ситуация иная, а именно, мы сначала определяем свои профессиональные педагогические задачи, а уже затем, в соответствии с ними, выбираем те или иные технические решения, в том числе тип компьютера, конфигурацию и т. п. Как видите, мышление и подходы к информатизации образования принципиально изменились.

При Московском департаменте образования создан экспертный совет, в который, в частности, входят руководители наиболее крупных организаций, работающих с образовательными учреждениями в области поставки компьютеров, компьютерных программ и технологий. Этот экспертный совет, рассматривая заявку, поданную тем или иным образовательным учреждением, обсуждает цели поставки компьютеров, профессиональные задачи, которые будут ре-

шаться, и т. д. Экспертный совет помогает подобрать руководителю правильную конфигурацию техники, программных и технических средств.

После оснащения образовательного учреждения встает проблема подготовки кадров. Нужно, чтобы кадры не только правильно понимали задачи, но и правильно их решали. Кстати, мы убедились на практике, что те организации, которые всерьез начинают работать с образовательными учреждениями, перестают быть просто посредниками или поставщиками, а, включившись в образовательный процесс и поняв его специфику, начинают модифицировать свои профессиональные интересы под решение специфических образовательных задач.

Мы убеждены в том, что залогом успеха при создании новых образовательных технологий является формирование работоспособных полипрофессиональных коллективов, состоящих из высококвалифицированных педагогов, ученых, инженеров, программистов и т. д. Только полипрофессиональные коллективы, включенные в образовательный процесс и решающие задачи образовательного процесса, способны разработать и реализовать принципиально новые образовательные технологии с использованием современных информационных технологий. Коммерческие же аспекты деятельности таких коллективов вполне возможны. Их создание — одно из стратегических направлений развития столичного образования.

Традиционно Московский департамент образования имеет хорошие международные связи. Ког-

да мы встречаемся с нашими зарубежными коллегами, то практически всегда касаемся вопросов, связанных с информатизацией образования.

Например, много интересного мы видели во время визита в Стокгольм. Мы договорились о совместной деятельности между двумя управлениями образования, в том числе и в области информатизации. В Швеции тоже имеются информационные центры, развита очень похожая система подготовки кадров, наши коллеги также занимаются созданием информационных сред для управленцев, учителей и учеников, причем в концепциях этих направлений у нас достаточно много общего.

Контакты с английскими и американскими коллегами показали, что с ними расхождений гораздо больше. Но это не мешает активно и со взаимным интересом сотрудничать, в частности, в создании образовательных мостов с использованием модемной связи для совместных школьных проектов. Существуют образовательные проекты и с Нидерландами.

Когда мы говорим о совместных телекоммуникационных проектах, то все согласны, что компьютер сокращает пространство и время, дает возможность детям общаться гораздо содержательнее, чем другие средства.

Есть еще интересный проект с Финляндией по созданию телекоммуникационного центра на базе одного из технических лицеев, в котором для участников будут предоставлены широкие возможности для общения в разных формах с использованием информационных технологий со странами Скандинавии. Кроме того, в этом

лицее будут готовить специалистов по телекоммуникациям, которых сейчас очень не хватает.

Московский департамент образования готовит международную конференцию «Образование без границ». Она заложена в программе «Столичное образование». Одна из секций этой конференции будет посвящена информатизации, как одному из наиболее значимых направлений современного образования, причем все более независимому от административного или политического деления карты мира.

Но надо отчетливо понимать, что реализация того, о чем говорилось выше, требует значительных бюджетных ресурсов. В этой связи я хотела бы сказать о понимании со стороны Московского правительства и мэра Москвы Юрия Михайловича Лужкова важности для развития Московского региона, а вместе с этим и для России в целом образования как отрасли, от состояния которой напрямую зависит наше будущее. Принятие программы «Столичное образование» — это подтверждение поддержки со стороны Московского правительства нашей стратегии по созданию новых образовательных технологий.

У Москвы, как столицы России, есть своя, особая роль, которая определена историческим и

культурным наследием, которая заложена в самом развитии Российского государства. С одной стороны, оно всегда строилось по принципу соборности, а это значит, что люди не разобщаются, а собираются. Православие наше также строилось. И в этом смысле Москва — центр притяжения. С другой стороны, в России всегда существовало некоторое противопоставление центра и периферии, но не в том смысле, что центр — это что-то хорошее, а периферия — плохое, а в том, что именно столица должна быть ответственна за все процессы, происходящие в государстве, и должна быть ведущей в этих процессах.

Мы живем в сложное время, в динамично развивающемся мире, вступаем, а по сути уже вступили в информационный век, и мне еще раз хочется повторить очевидное: от состояния нашей системы образования в значительной мере зависит статус России в мировом сообществе. Поэтому мы стремились, чтобы концепции, заложенные в программе «Столичное образование» и выбранные приоритеты были направлены на создание современного уровня системы образования Московского региона.

Московский департамент образования открыт для сотрудничества, и мы готовы к обсуждению наших концепций с коллегами.

## Уважаемые читатели!



Десять лет назад, когда началась информатизация образования, организация журнала была делом далеко не таким простым, как сегодня. Идея создания научно-методического журнала «Информатика и образование» принадлежит Александру Юрьевичу Уварову, бывшему в тот период начальником Управления информатики и вычислительной техники Министерства просвещения СССР. Только А. Ю. Уваров и мог «пробить» новый журнал через ЦК КПСС и другие организации. Главным редактором журнала в то время наиболее естественно мог быть Андрей Петрович Ершов —

отец советской школьной информатики, но он уже был редактором академического журнала «Программирование», а быть одновременно редактором двух журналов не полагалось. Поэтому мы с А. Ю. Уваровым предложили академику Владимиру Андреевичу Мельникову стать Главным редактором журнала «Информатика и образование». Академик В. А. Мельников был одним из основных участников создания самой замечательной отечественной машины БЭСМ-6. Владимир Андреевич сначала с некоторым сомнением, а потом с энтузиазмом согласился и во многом способствовал серьезному научному уровню журнала и связи его с компьютерным сообществом.

Я принимал участие в формировании первого и нескольких последующих номеров журнала. Поэтому мне особенно приятно, спустя десять лет, видеть, какое важное и серьезное место он занял в отечественной школе, и получить предложение сформировать «Московский выпуск» журнала.

Отличительным и важным, на мой взгляд, моментом настоящего времени является расширение прикладной информатики в рамках образовательной области технологии и ее вхождение в изучение других дисциплин.

Надеюсь, что читатели получат достаточно полное представление и о московской концепции информатики в школе, и о конкретных достижениях московских учителей, а настоящий выпуск послужит доброй памяти о двух выдающихся ученых — Владимире Андреевиче Мельникове и Андрее Петровиче Ершове.

А. Л. Семенов,  
доктор физико-математических наук,  
ректор Московского института повышения  
квалификации работников образования,  
заместитель Председателя  
Московского департамента образования

**А. Л. Семенов,**

*доктор физико-математических наук, ректор МИПКРО,  
заместитель председателя Московского департамента образования*

## ОБРАЗОВАНИЕ, ИНФОРМАТИКА, КОМПЬЮТЕРЫ

Обсуждение этой темы хочется начать с некоторых терминологических замечаний.

Начнем со слова «компьютеры». Говоря о компьютерах в контексте образования, мы часто имеем в виду нечто более широкое. Например, разговор может охватывать средства телекоммуникаций, не содержащие компьютеров. Но многие современные копируемые, стиральные машины, автомобили часто содержат микропроцессоры, которые могут работать, как универсальные компьютеры, однако стиральную машину компьютером не называют. Поэтому точнее было бы говорить о средствах новых информационных технологий (НИТ и СНИТ).

Второе замечание относится к слову «информатика». Обычно под этим словом понимается наука о переработке, хранении и передаче информации. Очевидно, важнейшим примером этих операций является человеческая деятельность, в частности мышление и общение. Поэтому при таком определении в область информатики немедленно попадают лингвистика, логика, философия. И в этом есть определенный смысл. В частности, математика человеческого мышления и коммуникации, открытая в течение последнего десятилетия прошлого века и первой трети нынешнего, нашла почти стопроцентное применение в современной информатике, столь занятой сейчас проблемами машинного мышления и коммуникации.

### Информатика

Мы упомянули о содержании информатики как науки. Что же понимается под информатикой в отечественной (и в известной нам зарубежной) школе? Обычно это понимание включает в себя элементы следующих областей:

- *математическая информатика;*
- *практические навыки работы со СНИТ, в частности умение писать программы на каком-либо распространенном универсальном языке программирования для компьютера;*
- *информационная культура, т. е. общее представление об информационных процессах в окружающем мире, об источниках той или иной информации, средствах массовой информации, системе морально-этических и юридических норм, ценностная ориентация.*

Рассмотрим эти области последовательно.

### Математическая информатика

Как было отмечено выше, эта область знания отнюдь не привязана накрепко к НИТ, это не наука о компьютерах и их использовании, это часть математики, пользующаяся общематематической системой понятий и методов. Ее предметом являются прежде всего конечные (конструктивные) объекты и конструктивные же (алгоритмические) операции над ними. Задачи, которые ставила перед собой математическая информатика, были в течение первого периода ее развития мотивированы проблемами человеческого мышления и коммуникации. В процессе изучения этих проблем были построены математические модели, которые легли в основу теоретических построений, относящихся к компьютерам. Важнейшие определения и теоремы математической информатики были найдены до появления компьютеров. К ним относится как, например, утверждение о существовании универсального вычислительного устройства, так и факт о невычислимости некоторых конструктивно определяемых функций. Это неудивительно, поскольку математическая информатика как раз и абстрагируется от того, какова материальная основа процесса переработки информации (в частности, делает это человек или полупроводниковый микропроцессор). В ходе развития вычислительной техники и практики возникли различные задачи, которые, не меняя основных понятий и методов науки, мотивировали получение тех или иных конкретных результатов.

В то же время математическая информатика представляет собой наиболее важную и естественную среду для формирования некоторых элементов информационной культуры и многих общих (внематематических, жизненных) мыслительных навыков. На это обстоятельство обращал внимание основоположник советской школьной информатики А. П. Ершов.



## Борьба с компьютером

Итак, компьютеры не необходимы при изучении математической информатики.

Классическое сравнение изучения информатики без компьютера с изучением плавания без воды и езды на велосипеде без велосипеда (которое автор слышал, в частности, от предпоследнего президента АН СССР) частично основано на недоразумении. Изучать компьютеры без компьютеров почти наверняка крайне неэффективно, но исходный советский курс информатики, слава богу, не был курсом изучения компьютеров, а являлся в основном именно курсом математической информатики. Не стоит изучать команды операционной системы MS DOS или текстового редактора Лексикон без компьютера, однако приспособить компьютер для более ясного восприятия учащимися понятия колмогоровской сложности или работы алгоритма сортировки слиянием не так просто. Полезны ли компьютеры при изучении математической информатики? Их полезность примерно так же спорна или бесспорна, как и полезность компьютера при изучении других разделов математики. Полезность эта базируется на том, что практическое функционирование компьютера (с программой, заданной учителем или учащимся) может служить иллюстрацией некоторого теоретического положения или конкретной реализацией теоретического построения. При этом, если в построении была допущена ошибка, она может наглядно проявиться при работе компьютера (а может, конечно, и не проявиться). Особенно полезным оказывается компьютер, если задачи, решаемые для усвоения общих теоретических положений, относятся к визуальной — графической среде, а также если результат работы компьютера и соединенных с ним устройств по каким-то причинам интересен учащимся и имеет смысл независимо от изучения математической информатики. Что касается наглядности, то она может быть столь же полезна при изучении математики или физики, когда учащийся видит на экране разветвление процесса, для которого им построено математическое описание.

Следует, однако, понимать, что при использовании несовременных компьютеров и систем программирования преодоление технических трудностей и усвоение условностей требуют весьма существенных усилий учащегося и учителя. Если при этом упомянутая выше наглядность и нематериальная мотивация отсутствуют, то работа с компьютером в классе оказывается уделом увлеченных программистов-учителей и единиц учащихся-хакеров, которые находят удовлетворение именно в преодо-

лении технологии. В этой ситуации лучше бы вынести эти занятия на факультатив, а остальным ученикам дать возможность разобратся в достаточно увлекательной математической информатике (иногда называемой еще математикой конечного), не погружаясь в ФОКАЛ и MS DOS.

Отложим на время это обсуждение, поскольку оно относится к общему вопросу о полезности использования компьютеров в школе вообще (в частности, при изучении различных предметов).

## Насколько нужно осваивать в школе практические навыки использования СНИТ?

Нужно ли учить детей пользоваться ксероксом, факс-машиной, видеокамерой, текстовым редактором, электронной таблицей, телекоммуникационным пакетом, базой данных, видеомонтажным оборудованием и т. д.? В какой-то степени это полезно. Однако включение в ту или иную форму образования того или иного навыка должно учитывать следующие обстоятельства (желательно индивидуально для каждого учащегося или категории учащихся):

- *какие образовательные цели, помимо самого технического навыка, могут быть реализованы после его освоения (например, умение печатать на машинке и пользоваться редактором текстов дает возможность эффективно достигать целей развития коммуникативных способностей учащихся);*
- *найдет ли этот навык немедленное применение (на школьных уроках и вне их);*
- *насколько этот навык будет полезен в будущем;*
- *насколько сложно его освоение.*

Будущая полезность навыка определяется следующими факторами:

- *ожидаемая распространенность того или иного вида информационной деятельности;*
- *необходимость и устойчивость конкретной технологии.*

Например, в течение ряда лет для применения электронной почты или даже для копирования текстового файла надо было знать некую магическую последовательность нажатий клавиш. В какой-то момент эта необходимость исчезла, работа с файлами, их копирование или передача по электронным сетям стали осуществляться в едином графическом интерфейсе (изобретенном когда-то для компьютера Macintosh). Вопрос о специальном изучении этих навыков отпал. Интерфейс Macintosh был создан 10 лет назад, после этого мы 8 лет учили школьников вещам, которые им не понадобятся в будущей деятельности.

## Полезное умение

На сегодняшний день можно с уверенностью говорить только об одном сколько-нибудь трудоемком в усвоении умении, овладение которым в рамках любого образовательного учреждения безусловно необходимо, — это умение быстро вводить в компьютер текст на родном языке. Оно возникло до компьютеров, и наиболее распространенной формой его является «слепа́я десятипальцевая машиннопись». Ей уже более 120 лет, а в современном российском варианте — более 100. Для информационной технологии срок вполне достаточный, и не видно, чтобы, исходя из динамики развития событий, этот навык стал бы ненужным в течение ближайших десятилетий. Освоение этого навыка требует (в зависимости от исходных данных учащегося и достигаемого уровня) 10—30 часов. Таким образом, затрата учебного времени довольно значительная, хотя и не катастрофическая. Конечно, есть и другие полезные навыки, например умение пользоваться телевизором или копировальной установкой (ксероксом). Поскольку на их освоение требуется значительно меньше времени (скорее десятки минут, чем часов), то и вопрос об их использовании в школе легче решить положительно, если они могут найти немедленное применение в жизни учащегося. Если же говорить о навыках, требующих более значительного времени на освоение, то такая затрата, как правило, не может быть оправдана применением в будущем (имея в виду уже упомянутую изменчивость технологии). Таким образом, вводить изучение тех или иных информационных технологий следует лишь тогда, когда они действительно будут необходимы в жизни учащегося, в частности при изучении информатики, истории, в издательской деятельности школы и т. д.

## Умение программировать на конкретном языке программирования

Как правило, с точки зрения общего среднего образования ценность программирования на конкретном языке сама по себе невелика и непропорциональна трудности его освоения. Важно, как умение программировать будет использоваться в школе. Если оно находит немедленное интенсивное применение в информатическом разделе курса математики — это аргумент в его пользу. Но, повторим еще раз, то или иное построение, определение, теорема, метод включаются в этот раздел не потому, что они имеют отношение к конкретной

системе программирования или модели компьютера, а потому, что составляют содержание фундаментального математического знания.

Большинство реальных языков программирования созданы не для школы. Как и для других сложных технических систем, их создание есть результат компромиссного учета различных факторов. При этом относительный вес этих факторов совсем иной, чем естественно ожидать в образовательном контексте. Поэтому закономерно появление специальных языков и систем программирования для образовательных применений. В России наиболее распространенными являются версии Лого, МикроМир (Е-практикум, КуМир и т. д.) и «Роботландия». В этих системах первые шаги делаются очень легко, активно используется визуализация. В той или иной степени эти системы обладают еще одним ценным качеством — они могут быть использованы для решения задач, интересных ребенку. Бесспорное первенство в этом отношении, как показывает опыт школ в десятках стран мира, имеет Лого.

## Информационная культура

Это, конечно, важнейшая часть общего образования человека в быстро меняющемся мире наряду со столь необходимой нам экологической культурой и заново выстраиваемой культурой экономических и социальных отношений. В области этих новых культур более ясна необходимость тех или иных целей, чем способы их достижения. Особенно ценной выглядит возможность создания в школе ситуаций, воспроизводящих те или иные ключевые проблемы информационной культуры. К этому вопросу мы вернемся, когда будем говорить о применении СНИТ в школе.

## Мир с компьютером

Итак, действительно ли СНИТ могут быть полезны в школе, и если да, как эту пользу из них извлечь?

Начнем с самого главного. Мы постоянно слышим и говорим о необходимости трансформации школы в некотором направлении, которое описывается и понимается по-разному, но все же в одном русле идей. Говорим ли мы о развивающем образовании, о переходе от «школы знаний» к «школе способностей», о реализации принципа «учить учиться», об умении решать новые плохо определенные задачи или о мыслительных навыках высшего порядка, о выработке творческого, а не репродуктивного знания, мы имеем в виду нечто общее. В то же время, когда речь заходит

о нашем собственном предмете, самым простым оказывается задать такую-то главу из учебника.

НИТ тем временем радикально меняют потребности общественного производства, управления, сферы услуг в тех или иных ЗУН, даваемых школой.

Приведем базовый пример: нужно ли на производстве уметь умножать столбиком? Не нужно. Реагирует ли на это школа? Да, но медленно. Использование калькуляторов на уроках физики давно не криминал. В результате столбиком дети умножают хуже, чем их родители. В то же время дети, как категория населения, с имеющимися в ее распоряжении СНИТ, умножают лучше, чем взрослые — со своими СНИТ.

Следующий пример. Нужно ли на производстве красиво и без помарок писать? Не нужно. Реагирует ли на это школа? Пока в очень малой степени. Но научить 4—7-летнего ребенка пользоваться основными базовыми функциями текстового редактора, включая расположение русских клавиш, можно гораздо быстрее, чем писать ручкой. Почему соответствующая трансформация не происходит? Видимо, причин здесь несколько. Самая очевидная — материальная: не хватает компьютеров (или денег на них). Но и в школах, где компьютеров действительно много, учителя начальной школы не очень-то спешат перевести существенную часть уроков русского языка в компьютерный класс. Но ведь начать сразу быстро (и более грамотно — об этом чуть позднее) писать — это то, что необходимо для обеспечения творческого самовыражения личности ребенка, для развития его творческих способностей, так как, используя компьютер, дети могут сопровождать свои сочинения и иллюстрацией, и мультипликацией, и музыкой и т. д.

Еще примеры. Системы автоматизированного проектирования (САПР) школьного уровня имеют очевидные преимущества перед школьным черчением. Как идет процесс их внедрения в курс черчения? Можно вспомнить приборно-компьютерные комплексы для изучения естественнонаучных дисциплин, компьютерные музыкальные системы и т. д.

### Критический порог

Список примеров можно продолжить, в него войдут неиспользуемые модемы (никак не подключим), ксероксы (сразу сломался), даже кодоскопы (кончились фломастеры, да и нет времени готовить пленки) и многое другое.

Ни в Москве, ни в России, ни в мире компьютеры не применяются широко в преподавании различных предметов в

школе (а тем самым не реализуется и цель подготовки граждан общества новой информационной культуры). В чем причины? Мы, естественно, будем их анализировать на примере Москвы. Первая — неготовность учителей. Вторая, связанная с первой, — существование некоего критического порога в использовании технологий. На первой причине, важнейшей, мы остановимся поподробнее дальше. Вторую, более технологическую, прокомментируем сейчас.

Компьютер в качестве машины для создания текстов, казалось бы, не отличается принципиально от пишущей машинки. Но только компьютер привел к реальному вытеснению всякого профессионального письма от руки машинным. В случае самих компьютеров ситуация повторяется — профессионалы начинают их использовать для своих целей еще когда они очень неудобны и не так уж эффективны. Но в какой-то момент начинается цепная реакция, в которую оказывается включенным экономический фактор — массовый продукт должен и может быть дешев. Современные компьютеры перешагнули границу массовой доступности по удобности обработки текстов с появлением Macintosh 10 лет назад. По телекоммуникационным возможностям это в основном произошло за последние три года. Сейчас они перешагивают границу доступности в области работы с мультимедиа информацией. Мультимедиа компьютеры, существующие около 10 лет, только-только начинают включать большую часть тех возможностей, которые мы естественно от них ожидаем. Наглядный пример — прием и запись обычных телевизионных программ на компьютер.

### Создание собственного информационного пространства

Возьмем две полярные ситуации:

- у человека нет под рукой ничего;
- у человека есть возможность доступа к любым информационным ресурсам.

В школе почти не учат поведению во второй ситуации, но именно она становится все более и более определяющей в нашей жизни. Сегодняшние школьники будут жить в мире, где любая информация будет столь же повсеместно доступна, как сегодня доступна электрическая энергия. Для того чтобы выйти на информационную магистраль, сейчас нужно иметь компьютер с телекоммуникационным интерфейсом (модемом). Для того чтобы адекватным образом предъявлять человеку найденную информацию (т. е. воспроизвести на исходном языке текст, сыграть музыкальное

произведение, продемонстрировать видеозапись), компьютер должен удовлетворять естественным требованиям для возможности воспроизведения информации. Эти требования обычно характеризуют так называемые мультимедиа машины, производимые сегодня практически всеми основными компьютерными компаниями. Проблема, однако, в том, что электрическая энергия однородна и наличие розетки в комнате решает проблему. В случае же информации наличие компьютера с модемом — это лишь самое начало. Понять, что искать, столкнувшись с той или иной жизненной ситуацией, и осуществлять процесс поиска — вот умения, которые становятся решающими.

### **Инструменты сообщения информации другим**

Вторая проблема — это умение сообщить то, что ты хочешь, другим людям. Возможность этого сейчас тоже меняется кардинальным образом. Сообщение, помещенное десятилетним школьником на доску объявлений в «Интернете», не менее доступно всему человечеству, чем сообщение с первой страницы «Нью-Йорк Таймс» или из вечернего выпуска «Вестей». Умению выражать свои мысли словами всегда учили в школе. Однако сейчас это умение может стать намного более важным. Создание мультимедиа сочинений и презентаций, гиперструктур и баз знаний, являющихся посланиями другим людям, — вот некоторые из основных направлений здесь.

### **Инструменты внутреннего запоминания информации**

Естественно предположить, что в ближайшем будущем сохранится ситуация, при которой людям будет нужно знать те или иные вещи наизусть. Но и здесь, вероятно, возникнут существенные изменения. Можно надеяться, что процесс выучивания будет происходить и более сознательно, и более мотивированно. Посмотрим на сегодняшнюю школу — по всем предметам время от времени задают что-то выучить наизусть, но не учат, как это делать эффективнее. Это является частным случаем проблемы, поставленной Сеймуром Пейпертом в двух его книгах. У нас нет даже названия для науки об умении учиться и нет места для этой науки в школе. Возможность использования СНИТ для запоминания той или иной информации изучалась достаточно широко и весьма популярна среди российских учителей. Мы, однако, подчеркиваем, что эту проблему не стоит рассматривать изолированно в рамках

какого-то фрагмента какого-то существующего школьного курса. Нам кажется, что речь стоит вести о глобальной попытке в рамках всех школьных курсов произвести сдвиг от ситуации воспроизведения некоего заученного знания к ситуации его использования (от «школы памяти» к «школе способностей»). Лозунг этот вызывает сочувствие достаточно давно, но НИТ делают его реализацию просто насущной необходимостью. В случае одного из самых сложных в этом отношении предметов — истории один из подходов, который становится все более и более популярным, — «задачник по истории». Другой подход (не противоречащий этому) — постоянное поддержание для каждого из учащихся более глубокого уровня знаний какого-то особого, специфического аспекта исторического процесса. Такое поддержание невозможно осуществить, базирясь только на школьном учебнике. Оно будет постоянно требовать привлечения дополнительных источников. Время от времени элементы таких индивидуальных информационных пространств должны, естественно, востребоваться, желательно не только учителем.

### **Инструменты добычи информации**

Поиск в электронных сетях и CD-ROM, конечно, не единственный способ получения информации об окружающем мире. Мы начали с него просто потому, что он является наиболее естественным продолжением книжной премудрости современной школы. Конечно, во взрослой жизни не менее важны наблюдение за окружающей природной и социальной средой, систематизация и анализ наблюдений. Следующей стадией здесь является создание условий, в которых становится возможным то или иное наблюдение — проведение научного эксперимента. СНИТ необыкновенно расширяют возможности сбора информации и постановки эксперимента. В руках учащегося оказываются средства от цифрового фотоаппарата до электрокардиографа и озонометра. При этом результаты измерений могут быть мгновенно проанализированы и представлены в наглядной форме.

### **Инструменты моделирования**

Все вышеприведенное дополняется, как мы надеемся, умением мыслить — строить и проверять обобщения и гипотезы, сопоставлять факты, делать умозаключения и т. д. Остановимся на одной из важнейших способностей этой категории — умении моделировать, проектировать и конструировать. В современной школе тематические модели действительности

широко распространены в учебных задачах математики (так называемые текстовые задачи), физики, химии. Современные СНИТ позволяют, если математическая модель уже задана, легко получить результаты моделирования (как правило, в числовом выражении, а если это в принципе возможно, то и в формульном). Человеку остается самая тонкая работа — построение математической модели, понимание области ее применимости, интерпретация результатов моделирования.

### Конструкторы

Конструирование — важнейшая деятельность человека, и нам кажется, что ее роль в образовательном процессе должна быть существенно больше. Разумеется, можно расширить понятие конструирования, включив в него, например, любое создание текста или любое построение человеческого знания. Этот подход может быть весьма плодотворен, на нем основана образовательная философия конструктивизма. Мы, однако, ограничимся более традиционным понятием конструирования — как создания какого-то устройства, объекта из элементов, блоков. Исходной ситуацией для нас будет детский конструктор, к которому мы все время будем обращаться. Разрешим себе сразу два обобщения: будем конструировать не только объекты, но и процессы, не только на столе, но и на экране компьютера. Между инструментами и конструкторами и даже между отдельными случаями их применения нет непреодолимой границы. Одна и та же ситуация может иногда рассматриваться и как применение инструмента, и как конструирование, так как в ходе конструирования мы используем некие инструменты, с помощью которых мы строим, конструируем некий объект. В детском конструкторе имеется паяльник или отвертка, а в инструментальном ящике — припой или метизы. Решающим, когда мы говорим об инструментах, является строящийся объект сам по себе. В ситуации конструирования этот объект обычно мыслится как модель чего-то.

Компьютер предоставляет в наше распоряжение неограниченный набор инструментов конструирования, все существующие информационные среды, все модели вещественных сред и некоторые очень интересные мостики во внекомпьютерную реальность. Отсюда необыкновенно широкие перспективы проектного метода обучения.

### Метод проектов

Вот начальные примеры простейших проектов, разрабатываемых и осуществ-

ляемых на базе компьютерных конструкторов:

- ребенок составляет произвольное слово из кубиков, компьютер его читает вслух;
- ребенок собирает из блоков программу управления движением робота в лабиринте или черепашки на плоскости, а компьютер демонстрирует это движение;
- ребенок собирает на столе из блоков компьютерного Лего модель реального или придуманного им устройства, собирает на экране подсоединенного к Лего компьютера программу управления, и спроектированный процесс развертывается в физическом пространстве и времени;
- ребенок собирает на экране фантастическую установку, состоящую из отдельных элементов с простыми функциями (например, двигатель может привести в движение транспортер, транспортер может переместить шар, шар, доехав до конца транспортера, — упасть и запустить следующий двигатель и т. д.), компьютер приводит экранную установку в действие;
- ребенок собирает физический прибор на экране или модель реального физического процесса вне лаборатории, компьютер демонстрирует на экране ход этого процесса;
- ребенок собирает физический прибор, компьютер получает информацию от датчиков в составе прибора, анализирует и отображает ее;
- ребенок задает на экране систему формул (в более сложных случаях блок-схему взаимодействия), включающую формулы, как в системной динамике), описывающих процесс, компьютер демонстрирует графически и численно ход процесса;
- ребенок проводит геометрическое построение на экране, компьютер сообщает результаты измерений в этом построении, повторяет то же построение при различных вариациях начальных условий;
- ребенок задает химическую формулу, компьютер дает трехмерное изображение молекулы;
- в созданной кем-то до него компьютерной модели экологической, экономической или социальной системы ребенок осуществляет действия, создает объекты и ситуации, принимает решения, компьютер развертывает перед ним результат этих действий и решений;
- используя имеющийся в компьютере каталог готовых элементов машин, ребенок строит модель машины на экране в терминах обычного машиностроительного черчения, компьютер изготавливает чертеж и моделирует на экране работу машины;
- ребенок «собирает» мелодию на экране из блоков, аранжирует ее и может медленно прослушать. Но это мы, кажется, уже вторгаемся в область, куда пока не хотели заходить.

Таким образом, решение многих из рассмотренных выше вопросов оказалось связанным с проблемами их применения учащимися в жизни, в частности в учебной деятельности.

**Л. Б. Переверзев,**

*заведующий лабораторией дизайна образования ИНТ*

## ПОЛЮБИТЬ МАШИНЫ, ПОМОГАЮЩИЕ УЧИТЬСЯ (Образовательная философия Сеймура Пейперта)

### Введение

#### «Лого» — только ли компьютерный курс?

Почти все отечественные школы, обладающие компьютерами типа IBM или Macintosh, располагают (или скоро будут располагать) также пакетами LogoWriter (ЛогоРайтер) или их новой, гораздо более мощной версией MicroWorlds (ЛогоМиры). Разработанные на базе оригинальной концепции образования, созданной профессором Сеймуром Пейпертом из Массачусетского технологического института и кратко именуемой «Лого», они представляют собой универсальные учебно-развивающие среды, т. е. открыты для занятий любым школьным предметом. С. Ф. Сопрунов — математик, программист и педагог, более всех других сделавший для распространения этих пакетов в России, — замечал в своей книге: «Можно предположить, что учитель астрономии предложит своим ученикам изобразить движение планет, учитель истории — проиллюстрировать жизнь средневекового замка, а учитель русского языка — формализовать грамматическое правило. Мы надеемся, что LogoWriter может быть использован в школе не только в виде специального «компьютерного» курса» [1].

Увы, подобная надежда покуда не сбылась. Несмотря на ряд воодушевляющих примеров, о чем хотелось бы поговорить особо, в большинстве наших школ оба пакета используются, как правило, лишь на занятиях информатики, при изучении основ программирования. Причиной тому — слабая осведомленность преподавателей других предметов о природе и образовательных возможностях Лого. Для знакомства с этой педагогической концепцией можно рекомендовать основополагающую монографию Пейперта «Mindstorms», впервые опубликованную на Западе полтора десятилетия тому назад, а позже вышедшую и на русском языке [2, 3].

### Нераскрытый потенциал

К сожалению, эту уже давно классическую работу, изданную в нашей стране сравнительно небольшим тиражом и сразу исчезнувшую с прилавков, довелось прочесть далеко не всем преподавателям информатики, не говоря уже об остальных. Да и среди прочитавших, судя по выборочным опросам, не каждый готов утверждать, что продумал и уяснил себе пейпертовскую концепцию достаточно всесторонне и глубоко, чтобы по достоинству ее оценить и уверенно определить к ней собственное отношение.

Между тем в 1993 г., тринадцать лет спустя после «Mindstorms», Пейперт опубликовал вторую книгу — Children's Machine. Rethinking School in the Age of Computer (Машина детей. Переосмысление школы в век компьютера), существенно дополняющую, а в чем-то и корректирующую первую. Пропагандируемые им и его соратниками идеи и практика образования на базе новых информационных технологий завоевывают все большее число последователей в учебных заведениях передовых стран мира. Движение Лого сделалось подлинно международным.

Сегодня в наших школах появляются учебно-развивающие среды Лого-Лого, благодаря которым чисто экранная компьютерная среда обретет третье измерение, т. е. станет объемной, осязаемо вещественной и телесной. Это необыкновенно расширяет горизонты познавательной, проектно-конструкторской и художественной деятельности детей, но учителям приносит множество совсем новых и невероятно сложных проблем, о которых стоит подумать загодя, чтобы успеть к ним подготовиться.

Короче, потенциал Лого велик, однако актуализирован пока в сравнительно малой степени. Оговоримся, что смысл и содержание этого движения отнюдь не сводятся целиком к имени и личности его лидера. Фактически там явно и неявно синтезируются общие идеи, изобретения и открытия таких выдающихся умов двадцатого столетия, как Л. С. Выгот-

ский. Жан Пиаже, Норберт Винер, Грей Уолтер, Марвин Минский, Уолли Фюрциг, Алан Кэй, и многих других видных психологов, математиков, кибернетиков и творческих педагогов. Отсюда оправданна попытка сжато очертить философскую основу и, так сказать, общеобразовательную направленность того, что можно кратко ради называть «провозвестием Лого» или (с учетом сделанных оговорок) «учением по Пейперту».

### Кардинальные вопросы

Наиболее простой и надежный путь — кратко резюмировать нужные нам аспекты двух вышеназванных книг Сеймура Пейперта. В обеих ведется повествование от первого лица, отражены персональные взгляды и субъективный опыт автора, иллюстрируются яркие эпизоды его биографии и профессиональной карьеры. На их страницах патриарх Лого вновь и вновь поднимает и под разными углами рассматривает одни и те же кардинально важные для него вопросы.

Вот главнейшие из них.

- *Человек рожден безостановочно учиться всю жизнь, учение обогащает и приносит радость. Почему же так много людей им тяготеют и от него отвращаются?*
- *Из чего складывается и как протекает столь успешный и многогранный «естественный» процесс учения ребенка от рождения до школы?*
- *В чем причина того, что после поступления в школу тяга и способность к учению у большинства детей резко падает, а подчас даже полностью прекращается?*
- *Почему уже в школе начинается губительный для учения раскол единой в своей основе культуры на две антагонистические сферы — естественнонаучную и гуманитарную?*
- *Устранимы ли препятствия, воздвигаемые перед учением нынешней школьной системой, за счет совершенствования ее устройства, и если нет, то что должно прийти ей на смену?*
- *Какие перспективы коренного обновления образования открываются с появлением компьютеров и каковы должны быть шаги к их практической реализации?*

Самим Пейпертом эти главные темы редко рассматриваются отдельно, а возникают сразу вместе. Я же попробую выделить и очертить их одну за другой

по возможности ближе к духу и букве авторского текста и потом вновь свести их в некое целое, добавив кое-какие комментарии от себя.

### Учение и школа

#### Матюфобия — боязнь учения

Любую учебно-образовательную проблему Пейперту свойственно рассматривать прежде всего в свете изучения математики — традиционно наиболее трудной, но вместе с тем наиболее универсальной и необходимой школьной дисциплины. Ведь, изучая математику, мы учимся оперировать основными категориями мышления: сравнивать, различать и отождествлять, логически рассуждать, делать умозаключения и доказывать истинность или ложность тех или иных высказываний. Однако у многих, особенно у людей, именуемых «гуманитариями», эта наука вызывает острую неприязнь, подчас подлинную «матюфобию». Главная вина здесь падает на школу, ибо она, «обрезая детей на ситуацию вынужденного учения, порождает у них ярко выраженное негативное отношение к математике, а возможно, и к учебе в целом» [2, 3].

Тут Пейперт указывает на примечательное лингвистическое родство: слово «математика» происходит из то же корня, что и древнегреческие *manthanein* (учиться), *mathema* (урок) и *mathematicos* (расположенный к учебе). Тем, кто боится математики, скорее всего просто не хватает решимости сделать над собой усилие, чтобы научиться упорядоченно и связно мыслить. По сути, такими людьми владеет страх перед учением как таковым. Не желая признавать этот уязвляющий их самолюбие факт, они предпочитают самое простое объяснение: отсутствие у них способностей к данному предмету. Сплошь и рядом (правда, по иным мотивам, связанным с нежеланием признавать дефекты своих методов) аналогичный приговор выносит им и школа.

Но допустим на минуту (хотя специальные исследования и конкретная практика экспериментальных школ убедительно доказывают обратное), что с математикой, физикой, историей, литературой или музыкой так оно и есть. Однако в случае с иностранным языком (чье положение в школе обычно столь же плачевно) здесь явно что-то не клеится. Труд-

но поверить, что поколения американских школьников, уже в раннем детстве беспрепятственно и без всякого специального обучения осваивающих английский, почти поголовно от рождения лишены способности к французскому, преподаваемому в школе. Во всяком случае подобная теория никак не объясняет, почему во Франции школьники, так же не испытывающие никакого труда с родным языком, столь же избирательно поражены неспособностью к английскому.

Скорее надо предположить, что школой в принципе не обеспечиваются какие-то фундаментальные условия, необходимые для полноценного усвоения каждым учеником не только математики и иностранных языков, но и всех других дисциплин. Поэтому тратить силы на какие-то частичные «улучшения» традиционной школьной системы не более разумно, чем пытаться усовершенствовать конный транспорт накануне изобретения двигателя внутреннего сгорания.

### **Математика, эвристическая игра и польза ошибочных теорий**

Математикой Пейперт называет «совокупность принципов, которым подчиняется учение». Он утверждает, что «как курс искусства учения... математика (под каким бы именем она ни приобрела известность) является для детей еще более важной областью изучения, нежели математика» [2, 3]. Прохождение такого курса нелишне и для взрослых. Многим из нас оно помогло бы ликвидировать фатальный страх перед учением, рассеять пагубное заблуждение относительно нашей природной способности или неспособности к определенным дисциплинам, увидеть глубинную связь между физико-математическими и гуманитарными науками.

Школа, однако, не спешит с решением этих первоочередных задач современного образования.

«Обычно считается хорошей практикой давать людям наставления и инструкции, касающиеся их главного занятия. Главным занятием детей является учение, думание, игра и тому подобное. Однако мы ничего не говорим им о такого рода вещах. Вместо этого мы рассказываем им о числах, грамматике и Французской революции, смутно надеясь, что из этой мешанины сложится нечто действительно важное как-то само собой. Иногда так получается. Но случаи отчуждения, ухода из школы и подростковой

наркомании не менее часты...» [4]. Знания о самом учении, т. е. знания о том, как следует учиться, Пейперт называет *математическими* знаниями.

Вторым важнейшим понятием, дополнительно проясняющим смысл математики, является *эвристика* — «искусство интеллектуального открытия», интуитивно-поисковый подход к решению проблем. Оба понятия заключают в себе принципы, «освещающие и облегчающие процесс учения». Процесс же этот возникает в любой ситуации приспособления к новому и овладения им; например, «когда некто сталкивается с новым устройством, новой танцевальной фигурой, новой идеей или новым словом. Сначала это новое соотносится с тем, что уже освоено или известно. Затем выясняется, в чем же состоит новизна и как с ней обращаться. Например, осваивая новое слово, мы сначала ищем знакомую «основу», а затем практикуемся в употреблении его в предложениях собственной конструкции» [2, 3].

Вместе с тем «новые знания часто противоречат старым, и эффективное учение предполагает стратегии, учитывающие такое противоречие» [2, 3]. Учащемуся нужно дать возможность не только работать с простыми и наглядными примерами, но также *играть* с ними и получать из опыта игры необходимые ему новые знания. При этом вовсе не обязательно (и даже нежелательно) идти по прямой линии «от одного истинного положения к другому, более совершенному». Естественный для детей «путь учения включает «ошибочные теории»: сравнивая их между собой и подвергая логической и экспериментально-эмпирической проверке, они научаются распознавать истинные и неистинные моменты, содержащиеся в любой построенной ими теории.

К сожалению, среда обычного школьного урока, где всегда требуется «правильный» ответ, наказывает любую ошибку снижением отметки, неудовольствием учителя или насмешками со стороны класса и тем заведомо исключает подобный путь. Нельзя ли исправить столь очевидно контрпродуктивный метод? Почему бы специально не организовать для учащихся «такую интеллектуальную среду, в которой критерии истинности и ложности не занимали бы доминирующего положения?» [2, 3].

Созданию такого рода эвристически-игровой, учебно-развивающей, в бук-



вальном смысле *общеобразовательной* среды и посвящена деятельность Пейперта на протяжении последней трети двадцатого столетия.

### Естественная среда учения

Проработав около пяти лет под руководством знаменитого швейцарского психолога Жана Пиаже, Пейперт заимствует у него «образ ребенка как зодчего, возводящего структуры собственного интеллекта».

Едва родившись, ребенок начинает учиться: на втором году жизни он «научается» ходить, к началу третьего — говорить и пользоваться ложкой, в четыре года он уже сам одевается и т. д. К поступлению в школу у детей обычно накапливается также неплохой запас интуитивных физико-математических знаний, в рациональной форме преподаваемых лишь ученикам средних и старших классов. Так, дети отлично оценивают размеры и другие количественные соотношения непосредственно окружающих их объектов. Например, они очень быстро и достаточно точно вычисляют расстояния и углы, скорость и ускорение, разложение сил и взаимодействие движущихся масс, когда бегают друг за другом, забивают футбольные мячи, фехтуют на палках или лавируют между автомобилями на улице.

Исходя из этого, Пейперт делает чрезвычайно важный для его концепции вывод: «Центральная математическая мораль такова: мы продемонстрировали, что научились выполнять нечто математическое без всякой инструкции — и даже несмотря на то, что нас учили делать это иначе» [4].

Тот факт, что рациональное освоение тех же знаний в школе для многих превращается в непреодолимое препятствие, приводит к еще одному, математически столь же важному, выводу. Есть «форма познания и учения, данная людям как бы от природы, но идущая наперекор всему, происходящему в Школе» [4]. Разрешим ли конфликт «естественного» и «школьного» учений и что именно придает последнему столь «противоестественную» направленность?

### Школа, превратившаяся в машину

В книге [2] Пейперт утверждал: «Если за образец «успешного учения брать способ, каким ребенок овладевает речью», то нужно считать «класс искуст-

венной и неэффективной учебной средой, которая была создана обществом, поскольку естественное окружение плохо соответствовало столь важным областям», как «школьные правописание, грамматика или математика». Однако есть шанс преобразовать учебную среду вне класса так, что «многие, если не все, школьные предметы, которым в настоящее время обучают с таким трудом, такими затратами и с таким ничтожным успехом, будут осваиваться так же, как дети учатся говорить, — легко, успешно и вне формально организованной сферы обучения. Это очевидно означает, что школ в том виде, в каком мы их знаем сейчас, в будущем не будет. Но вопрос о том, будут ли школы сами преобразовывать себя или же они отомрут и будут заменены чем-то иным, остается открытым» [2, 3].

Знакомство со школами разных стран мира привело Пейперта к более определенному суждению, в глазах многих крайне парадоксальному. Коротко оно звучит так: «Школа, пораженная болезнью прогрессирующей *искусственности* и *техникализации* («отехничивания»), уже давным-давно превратилась в гигантскую машину, основанную на безнадежно устаревших технологических принципах и неуклонно истребляющую в себе все живое; единственным средством успешного излечения школы от этого смертельного недуга может (при известных условиях) стать *новая технология*» [4].

Вот как раскрывается этот тезис: *наиболее естественен* «тот вид учения, который имеет место при здоровых взаимоотношениях между матерью и ее ребенком или между двумя знакомящимися друг с другом людьми. Но обучение в школе не является природным актом. Напротив, институт школы с ежедневным расписанием, фиксированным учебным планом, стандартизованными тестами и прочими атрибутами постоянно стремится редуцировать учение к серии технических операций, а учителей — к техникам (т. е. специалистам, имеющим дело только с машиной. — Л. П.). Конечно, это никогда не удается полностью, ибо учителя противятся навязываемой им роли техников и привносят теплые, естественные человеческие отношения в классные комнаты» [4]. Фактически «учитель занимает ту или иную промежуточную точку оси, на одном конце которой находится техник, а на противоположном тот, кого позволительно назвать подлинным учителем»

[4]. От того, какой из двух названных ролей и позиций будет отдано решающее предпочтение, зависит исход любой образовательной реформы, и выбирать здесь — учителям.

### Сбросить скорлупу технизированности

Медлить с подобным выбором уже невозможно. Главнейшая характеристика нашей эпохи — стремительно нарастающая скорость и масштабы *мегаперем*ен, затрагивающих буквально все аспекты современной действительности. Считать целью школьного образования механическую передачу детям пресловутой «суммы знаний», устаревающих быстрее, чем переписываются учебники и пересматриваются учебные планы, — абсурдно.

«Инструкционизм», т. е. подход к обучению, ставящий во главу угла рецептурные инструкции по поводу того, что ученику надлежит запоминать, а потом воспроизводить, должен уступить место «конструкционизму» — активному построению, *конструированию* самим учеником нужных ему понятий и знаний из имеющихся вокруг него элементов учебного содержания.

К тезису Пиаже о том, что каждый ребенок есть «зодчий, возводящий структуры собственного интеллекта», Пейперт делает кардинально важное дополнение: «Одной лишь свободы от внешнего принуждения для подобного рода «зодчества» еще недостаточно. Необходимо также *адекватное культурное окружение*, в котором ребенку было бы интересно учиться и самостоятельно добывать знания, которые позволяли бы ему справляться с нетривиальными и совершенно новыми проблемами, не имеющими аналогов в прошлом. Это культурное окружение должно быть таким, чтобы ребенок всегда мог найти там живые, яркие, лично близкие ему ассоциации со стоящей перед ним учебной задачей».

Бессмысленно ставить подобную задачу перед внутренне отехниченной школой, где царит «модель механического заучивания, когда материал воспринимается как лишенный смысла; иными словами... модель, лишенная ассоциаций» [2, 3]. Чудовищно инерционная, школа упорно противится каким-либо изменениям основ ее машинной структуры и принципиально не способна готовить учеников к полноценной деятель-

ности даже в сегодняшнем, не говоря уж о завтрашнем мире.

«Основной вопрос о будущем образования — будет ли технология укреплять или подрывать отехниченность того, что стало теоретической моделью и в очень большой степени реальностью школы» [2, 3]. Придать ей способность не только реагировать на мегаперемены, но и до какой-то степени предопределять их будущий курс, можно лишь одним путем: «используя технические средства для того, чтобы высвободить учение в школе из сковывающей его технической скорлупы» [4].

### Возникновение интеллектуальных моделей

#### От зубчатых колес к абстрактным идеям

Тему «технических средств», возвращающих учению «естественный» характер, Пейперт начинает с первой же строки «Mindstorms»: «Мне не было и двух лет, когда у меня возник интерес к автомобилям. Названия деталей машины составляли существенную часть моего словаря тех лет» и «предметом особой гордости было мое знание слова «дифференциал». Вскоре «игра с шестернями стала моим любимым занятием. Я был просто влюблен во вращающиеся наподобие шестерней круглые предметы... и прежде всего... собрал из «конструктора» простейшую систему зубчатой передачи».

Автомобилями увлекаются миллионы мальчиков, но не каждому из них приходится в голову, как это случилось с юным Сеймуром, «мысленно представлять вращающиеся колесики и выстраивать причинно-следственные цепочки» [2, 3].

Много лет спустя, читая труды Пиаже, Пейперт осознал свою увлеченность передаточными механизмами в терминах *ассимиляции* (здесь — уподобления какого-то неизвестного и чуждого нам объекта чему-то знакомому и своему. — Л. П.) и констатировал: «Любая вещь дается легко, если вам удастся ассимилировать ее в совокупности собственных моделей. Когда же этого не получается, то что угодно может оказаться мучительно трудным... Понимание учения должно быть *генетичным*, т. е. оно должно опираться на генезис (процесс зарождения

и становления) знания. Чему может научиться индивид и как он будет учиться, зависит от того, какими моделями он овладел. Это, в свою очередь, порождает вопрос, как индивид научился своим моделям. Следовательно, в «законах учения» должно раскрываться, как интеллектуальные структуры вырастают одна из другой, как в ходе этого процесса они приобретают не только логическую, но и эмоциональную форму» [2, 3].

### **Аффективная сторона и материалы интеллектуального зодчества**

Живое и взволнованное переживание ребенком рождающегося у него знания — таков первый из двух важнейших вкладов Пейперта в развитие тезиса о «зодчем собственного интеллекта», выдвинутого Пиаже. Последний намеренно, во имя строгой «научности», ограничивался рассмотрением только когнитивной, логически-познавательной стороны учения. Пейперт уделяет равное внимание стороне *аффективной*, связанной с переживанием индивидом тех или иных эмоций по поводу предмета и самого процесса учебы: «Мой контакт с передаточными механизмами был наполнен чувством, любовью, а не только пониманием» [2, 3].

Указанные механизмы служили *переходным объектом* от конкретных телесных манипуляций с осязаемыми вещами к построению абстрактных логико-математических понятий, и выбор для этой цели именно зубчатых колес был глубоко индивидуален, так как бессмысленно было бы «тиражировать» тот же самый переходной объект, «чтобы каждый ребенок смог приобрести тот же опыт». — «надеяться на это означало бы не понять самой сути моей истории. *Ведь я полюбил передаточные механизмы!* А это невозможно описать в чисто когнитивных терминах. Со мной произошло что-то личное, и нельзя утверждать, что с другими произойдет то же самое» [2, 3].

Второй вклад состоит в указании на то, что сама способность маленького зодчего возводить интеллектуальные постройки все большей сложности, объема и совершенства, равно как и возраст, в котором она проявляется, существенно зависит от того, какие «строительные материалы» при этом окажутся у него перед глазами. Приходят же они из ближайшего культурного окружения, весьма различающегося по составу и объему. «В

одних случаях культура поставляет их в изобилии... Сам факт, что многие из предметов обихода (ножи и вилки, мамы и папы, ботинки и носки) соотносятся друг с другом, становится «материалом» для построения интуитивного понятия о числе» [2, 3].

Но нельзя сказать того же о выработке классификационных понятий, ибо «наша культура относительно бедна на модели операций по систематизации». Поэтому «в большинстве случаев, когда Пиаже объясняет более медленное формирование отдельного понятия его большей сложностью и абстрактностью, я усматриваю решающие факторы в относительно плохой представленности в культуре тех материалов, которые могли бы превратить данное понятие в простое и доступное» [2, 3]. Напомним, что Пиаже проводил свои наблюдения над самостоятельной познавательной активностью детей, давая им (опять-таки ради чистоты полностью контролируемого эксперимента) лишь минимальный набор простейших предметов.

### **Технический Протей**

Но тут читатель, уже предупрежденный Пейпертом о бессмысленности «тиражирования» его любимых передаточных механизмов, которые для других детей, вполне вероятно, окажутся ничего не значащим куском железа, спросит: «Какова же тогда практическая ценность выводов, сделанных из события, носящего сугубо индивидуальный и единичный характер?»

В самом деле, культурой накоплено огромное количество вещей: орудий, механизмов, приборов, машин; промышленных и коммуникационных систем; стихотворений, музыкальных пьес, скульптур, живописных полотен, театральных спектаклей, песен, танцев и т. д. Не исключено, что любой артефакт (как, разумеется, и факт природы) может вызвать интерес и любовь у того или иного ребенка и послужить ему «переходным объектом», помогающим ассоциативно овладеть интеллектуальными операциями высокого уровня. Но как это выяснить? Предъявить наглядно и в доступной для манипуляции форме все предметное богатство культуры (а заодно и натуры) каждому двух-трехлетнему малышу или хотя бы первокласснику с тем, чтобы он все самостоятельно перепробовал и выбрал что-нибудь себе по вкусу?

Пейперт убежден, что прогресс вычислительной техники делает подобную идею не столь уж фантастичной. «Что не под силу передаточным механизмам, с тем может справиться компьютер», этот «технический Протей (древнегреческое божество-оборотень, непрестанно меняющее свой облик. — Л. П.), Его сущность — в универсальности, способности к имитации. Поскольку он может принимать тысячи ликов и выполнять тысячи функций, он может удовлетворить тысячи вкусов». Благодаря этому «использование компьютеров может изменить современные представления о возрасте, в каком ребенок может нечто понять», и «поставить под сомнение общепринятые постулаты возрастной и дифференциальной психологии, а также наши представления о формировании установок» [2, 3].

Однако для этого нужно прежде всего радикально изменить исходное понимание роли компьютера в образовании. По широко распространенному мнению, компьютер используется для того, чтобы «обучать ребенка», или даже «программировать ребенка», который при этом трактуется как некий автомат для выполнения заранее предопределенных операций и достижения целей, поставленных ему кем-то другим. Тем самым «продолжается насильственное насаждение трудноусваиваемого материала, доставшегося в наследство от докомпьютерной эпохи» [2, 3].

Позиция Пейперта диаметрально противоположна: «ребенок программирует компьютер и, делая так ...не только овладевает частичкой самой современной техники, но и приобщается к некоторым из самых глубоких идей естествознания, математики, а также к искусству интеллектуального моделирования» [2, 3].

К такого рода взглядам создатель Лого пришел, по его словам, в стремлении «превратить компьютер в инструмент настолько гибкий, чтобы многие дети сумели создать для себя нечто похожее на передаточные механизмы моего детства» [2, 3]. Обосновывая эту идею, Пейперт описывает ситуацию, когда ребенку предлагает сперва рассмотреть в качестве переходного объекта учения *собственное тело*, обладающее не только природными, но и до некоторой степени культурными чертами.

### **Вовлечь собственное тело в познание и рефлексировать о нем**

Сперва надо «дать детям способ размышлять о себе как о «творящих науку», когда они занимаются вещами, доставляющими им физическую радость», например шаганием на ходулях или жонглированием. Это позволит сделать для них очевидным «тот факт, что овладение физическим навыком во многом схоже с построением научной теории» [2, 3].

Как известно, некоторые знаменитые спортивные тренеры «затрачивают немало усилий на анализ и вербализацию движений, которым должны выучиться и которыми овладевают их подопечные» для достижения рекордных результатов. Т. Галлуей, автор книги «Внутренний теннис», «побуждал своих учеников думать так, как если бы они состояли из двух «я»: одно из них — аналитик, склонный все описывать словами, а другое — более целостная, склонная к интуитивному восприятию натуры», причем «оба «я» должны контролировать друг друга». Очень важную часть выработки физического навыка составляет при этом «обучение тому, какому из двух «я» отдать предпочтение в тот или иной момент» [2, 3].

Предлагая тот же подход ученику, мы относимся к нему «как к эпистемологу», т. е. исследователю феномена познания: «ребенок поощряется в стремлении стать знатоком в распознавании и выборе различных способов мышления», помогающих приобрести нужный телесный навык. Но необходимость осуществлять подобного рода рефлекссию (рефлексировать — значит мыслить о собственном мышлении) порождает специфические проблемы языкового плана. Обыденный «современный язык недостаточно богат, чтобы описать эту область деятельности», поэтому очень кстати оказываются более формализованные и структурированные способы рассуждения, заимствованные из дискретной алгебры и теории алгоритмов.

Взглянем, не вдаваясь в подробности, на две соответственно построенные стратегии обучения процессу жонглирования, где первая отражает ход мыслительного процесса, характерного для языка «наивного» наблюдателя, а вторая — более вдумчивый и аналитичный рефлексивный метод.

## Теоретическая тренировка в жонглировании

Первая стратегия. Если просто посмотреть на движение мячей в руках умелого жонглера, то можно составить следующее описание необходимых для тренировки «биомеханических» действий:

1. Начало. Мячи 1 и 2 находятся в левой руке, мяч 3 — в правой.
2. Перебрасываем мяч 1 по крутой параболе в правую руку.
3. Когда мяч 1 достигает вершины параболы, бросаем мяч 3 из левой руки по такой же крутой параболе, но так, чтобы траектория мяча 3 проходила под траекторией мяча 1.

4. Когда мяч 1 достигает правой руки, а мяч 3 находится в вершине параболы, перебрасываем мяч 2 по траектории, проходящей ниже траектории мяча 3. И т.д.

Приведенное выше можно было бы кратко записать в виде серии процедур типа БРОСОК ПРАВОЙ, БРОСОК ЛЕВОЙ и т. д. Однако эта примитивно-линейная инструкция, лишенная сколько-нибудь глубокого анализа и обдумывания событий, составляющих суть жонглирования, малопригодна для овладения навыком (разве что ценой огромного числа попыток с медленным и случайным нащупыванием вслепую правильного пути).

Вторая стратегия. По сравнению с первой она предполагает не только наблюдение и подражание, но и *размышление* над смыслом происходящего с опорой на достаточно развитые аналитико-синтетические процедуры. Принципиально важно то, что в программном плане деятельность жонглера перестает быть чисто последовательным процессом смены одной операции другой; теперь она складывается из нескольких *паралельных* процессов.

Для описания соответствующих процедур вводится новый элемент программирования: понятие «ДЕМОНЫ КОГДА». Оно поясняется инструкцией: когда наступает определенное событие, надо произвести определенное действие. Метафора «демон» отражает ту идею, что команда подается неким автономным устройством — специальным навыком, сформированным у человека, или отдельной вспомогательной программой, вложенной в компьютер. До поры это устройство бездействует, но в нужный момент оно, подобно демону, бросается

выполнять свою единственную функцию.

Жонглирование использует двух ДЕМОНОВ КОГДА:

КОГДА что-нибудь БРОСОК ЛЕВОЙ,

КОГДА что-нибудь БРОСОК ПРАВОЙ.

Конкретно это обозначает:

КОГДА ВЕРШИНА ПРАВОЙ БРОСОК ПРАВОЙ

КОГДА ВЕРШИНА ЛЕВОЙ БРОСОК ЛЕВОЙ.

При этом мы, естественно, допускаем, что обучаемый умеет выполнять названные броски и распознавать нужные состояния (положения мяча), а также объединять эти умения в процедуру ВЫПОЛНЯЕМ ЖОНГЛИРОВАНИЕ.

Посмотрим теперь, как эти теоретические построения переводятся в учебную практику

Шаг 1. Проверяем, умеет ли обучаемый бросать. Даем ему один мяч и просим перебросить из одной руки в другую. Обычно это действие выполняется легко, но часто требуется его совершенствование. Стихийно выполняемая процедура содержит ошибку.

Шаг 2. Проверяем, что учащийся умеет совершать броски. Даем два мяча и следующие инструкции:

ПЕРЕБРАСЫВАЕМ

БРОСОК ЛЕВОЙ

КОГДА ВЕРШИНА ПРАВОЙ БРОСОК ПРАВОЙ

КОНЕЦ

Это направленный обмен мячами между левой и правой руками. Хотя кажется простой комбинацией БРОСОК ЛЕВОЙ и БРОСОК ПРАВОЙ, обычно сразу же это действие не получается.

ШАГ 3. Ищем ошибку в процедурах броска. Почему не срабатывает процедура ПЕРЕБРАСЫВАЕМ? Обычно обнаруживается, что учащийся бросает мяч не так хорошо, как это казалось на шаге 1...

ШАГ 4. Отладка. Допустим, что ошибка заключалась в прослеживании глазами за полетом первого мяча, мы отладим этот промах учащихся, вернувшись к бросанию одного мяча, и т. д.

Как видим, вторая стратегия, составленная на основе логических процедур структурного программирования, отличается от первой тремя очень важными моментами. Она включает проверку ряда необходимых условий, допускает возможность *ошибок* и предусматривает *операции по их исправлению*. Ясно, что она

позволяет выработать нужный навык гораздо быстрее и экономнее. Но главное, чему учится здесь ребенок, — не жонглирование само по себе (хотя и оно полезно для развития наблюдательности, собранности и ловкости). Улучшая стратегию тренировки в подбрасывании и ловле шаров, он приобретает умение учиться чему угодно посредством сознательного критически-рефлексивного контроля за действиями своего тела при четком различении и анализе всех значимых особенностей поведения этого «переходного объекта» [2, 3].

Оговоримся, что не совсем верно называть «переходным объектом» тело само по себе. В рассмотренном случае речь шла о теле жонглера, но тем же математическим целям могло бы послужить тело теннисиста, человека, передвигающегося на ходулях и вообще занимающегося любой предметной деятельностью с использованием любых материалов и орудий.

### Объект с чертами субъекта

Тенденция трактовать собственное тело как универсальное моделирующее устройство, помогающее пониманию того, «как люди мыслят и как они учатся этому», проходит красной нитью в обеих книгах Пейперта. Она прямо вытекает из его детской любви к зубчатым колесам, с которыми он «телесно» отождествлялся в своем воображении.

«Думая о передаточных механизмах, я мог пользоваться собственным телом. Я мог почувствовать, как вращаются шестерни, воображая, что это вращается мое тело, я как бы вовлек собственное тело в познание того, как размышлять о передаточных механизмах. И наконец, поскольку, в сущности, связь между шестернями во многом определялась математическими знаниями, она позволила мне осознать передаточные механизмы как формальные системы. Я описал способ, каким передаточные механизмы послужили мне «объектом, с помощью которого думаю». Они стали для меня таким объектом, который помог мне превратиться в математика. Эти же передаточные механизмы послужили мне объектом, с помощью которого я стал осмысливать свою работу как исследователя в области образования. Моя цель — придумать другие такие объекты, которые дети могли бы осваивать сами, превращая их в свои способы осмысления мира» [2, 3].

В поисках «объектов, объединяющих

в себе культуру, знание и возможность личностной идентификации», была придумана *черепашка* — управляемый посредством компьютера кибернетический механизм, имитирующий (предельно грубо, но максимально доходчиво) такие свойства живого организма, как подвижность, чувствительность и память.

Но замечательнее всего то, что, в отличие от живых черепах (и даже выстихших приматов), это искусственное «существо» способно «понимать» и неукоснительно выполнять определенные поручения, отданные ему в письменном виде на языке Лого, очень близком к разговорному человеческому языку! Тем самым оно представляет собой не просто еще один объект для манипулирования, хотя бы и пробуждающий множество ассоциаций, полезных в процессе учения, но нечто значительно большее.

В образе черепашки навстречу ребенку выходит соучастник в играх, партнер по диалогу и взаимодействию, «субъект», имеющий имя и лицо, приглашающий увидеть в нем отражение самого себя — свое «второе я».

### Черепашка, которой приятно командовать

Известны две основные разновидности черепашки — напольная и дисплейная. Обе выполняют сходные математические функции, но отличаются друг от друга тем, что у первой есть вещественное «тело» из пластмассы и металла, а у второй осязаемого тела нет и она представлена чисто оптически — фигуркой на экране монитора. Напольная, как видно из ее названия, передвигается по полу или иной гладкой поверхности — всегда «головой» вперед и прямолинейно — на скрытой под ее сферическим панцирем тележке с электромоторчиками. Она снабжена пером, пребывающим в двух положениях — поднятым или опущенным. Опущенное перо оставляет за собой след на лежащем внизу листе бумаги и вычерчивает траекторию черепашкиного движения. Дисплейная черепашка, выступающая в аналогичной роли чертежника (и могущая по вашему указанию изменять свой цвет), продельывает то же самое на площади экрана, и далее мы будем подразумевать в основном именно ее.

Как уже было сказано, черепашка «понимает» и «исполняет» обращенные к ней «приказы», или *команды*, для чего малышам нужно нажимать специальные

клавиши, а детям постарше — набирать определенные слова и диффы на клавиатуре компьютера. Элементарные команды языка Лого передвигают черепашку по прямой линии вперед или назад на заданное число шагов; поворачивают вокруг своей оси вправо или влево на заданное число градусов; поднимают или опускают перо. Цепочка из нескольких элементарных команд задает программе более сложного поведения кибернетического механизма.

Команда НАПРАВО 90 заставляет черепашку развернуться вокруг своей оси на  $90^\circ$  (вы можете указать и любое иное число градусов). Команда ВПЕРЕД 100 — продвинуть по прямой на 100 шагов. ПЕРО ОПУСТИТЬ заставляет черепашку опустить перо, а перо ПОДНЯТЬ — поднять его. Кроме того, в «черепашном языке» имеется команда ПОВТОРИТЬ.

Допустив, что в исходном положении черепашка являет собою точку, не так уж трудно придумать несколько отдаваемых одну за другой элементарных команд, исполняя которые черепашка вычерчивает пером квадрат и треугольник. «Поскольку изучение управления черепашкой похоже на изучение родного языка, оно мобилизует эрудицию ребенка... запоминание ребенком команд, и он получает удовольствие от этого. Чтобы заставить черепашку двигаться по квадрату, вы сами проделываете этот путь, а затем описываете на языке «Разговор с черепашкой», как это у вас получается» [2, 3].

### Идея о плодотворных идеях

На первых порах детям легче поступать так: набрать сначала только одну элементарную команду — скажем, для одной стороны квадрата — и посмотреть, что выйдет; увидев отрезок прямой, проведенный черепашкой-чертежником, дать ей команду поворота; опять посмотреть на полученный результат, набрать команду для проведения второй стороны и т. д. Но вскоре они научаются сразу же составлять цепочку всех команд, нужных для получения квадрата, т. е. составляют целую программу (буквально — *преднамерение*) построения квадрата.

Далее дети обнаруживают: уже составленной программе для вычерчивания квадрата можно присвоить имя КВАДРАТ или KB, а программе для треугольника — ТРЕУГОЛЬНИК или TP, поместить их в «память» черепашки и тем «научить» ее сразу же запускать про-

граммы вычерчивания соответствующих фигур в ответ на эти короткие команды-имена.

Однажды придуманные и хранящиеся в памяти программы можно использовать как *могули* для создания еще более сложных программ, например заставляющих черепашку нарисовать домик, соединив вместе квадрат и треугольник.

Но вот ребенок спрашивает: «Как мне сделать так, чтобы черепашка нарисовала круг, колесо, обруч?» Путь поисков уже известен: попробовать самому совершить требуемые движения, вообразив себя черепашкой, передвигающейся только по прямой. Тут само тело подсказывает решение: один маленький шагок вперед, чуть повернуться направо, еще такой же шагок, опять чуть повернуться, еще шагок и так далее. Пишем две элементарные команды черепашке: ВПЕРЕД 1 НАПРАВО 1. Но программа из нескольких десятков одних и тех же элементарных команд выглядит слишком громоздко, и рано или поздно дети находят более изящное решение: ЭТО КРУГ ПОВТОРИ 1000 [ВПЕРЕД 1 НАПРАВО 1] КОНЕЦ.

Круг, правда, получается не совсем «круглым», но зато есть хороший повод коснуться интригующего вопроса о противоположности непрерывного и дискретного, «квадратуре круга», о методе последовательных приближений, о представлении окружности в виде многоугольника с числом вершин, стремящихся к бесконечности. После этого не так уж трудно побудить большинство детей самостоятельно сформулировать «Теорему о замкнутом пути черепашки»: «Если черепашка продвигается по границе какого-нибудь участка и заканчивает свой путь в том же состоянии, в котором она начала его, то сумма всех ее поворотов составит  $360^\circ$ ».

Экспериментально-конструктивное доказательство подобного утверждения дает несравненно больше, нежели «инструктивное» заучивание того раздела учебника, где говорится, что «Сумма внутренних углов треугольника равна  $180^\circ$ » (или «сумма внешних углов многоугольника равна  $360^\circ$ »).

Во-первых, данная теорема «более плодотворна»: ребенок на самом деле может ею пользоваться в той деятельности, которая его интересует. Во-вторых, она более *обобщена*: она применима не только к треугольникам, но также и к квадратам, и к кривым. В-третьих, она более *доступ-*

на: ее смысл легче ухватить. И наконец, она более личностна: вы можете «прошагать» эту теорему, и этот способ моделирования вырабатывает привычку связывать математику с собственными знаниями». Но самым важным становится понимание того, что «наиболее плодотворной из всех является идея о плодотворных идеях» [2, 3].

### Механизм, подобный организму

К числу плодотворных идей, возникающих при занятиях с черепашкой, принадлежит мысль о создании автомата, выполняющего заданную команду не «слепо», но варьирующего ее, смотря по обстоятельствам. Такой автомат должен воспринимать изменения окружающих условий, соответственно на них реагировать и так или иначе к ним адаптироваться.

Как придать черепашке хотя бы самое грубое подобие того адаптивного поведения, которое проявляет животное, когда оно пытается выбраться из тупика путем многократных проб и ошибок? Небольшое размышление приводит к ответу: надо предварительно усложнить устройство кибернетического механизма, снабдив его простейшим аналогом органа осязания. У напольной черепашки таким «органом» служит датчик касания с двумя электродами, замыкающимися при столкновении с любой вещественной преградой. Детям очень нравится сочинять программы, выполняя которые черепашка будет «вести себя» наподобие живой черепахи, оказавшейся в сходной ситуации.

Например, встретив на своем пути шкаф, черепашка отойдет назад, повернется на 90° и начнет движение вдоль его стенки, периодически поворачиваясь к ней «головой» и «пробуя», не исчезло ли это препятствие, а обнаружив, что его больше нет, тут же завернет за угол шкафа и будет двигаться опять в первоначальном направлении. Можно задать и такую программу, чтобы черепашка обходила весь шкаф по периметру один раз, многократно или бесконечно (во всяком случае, пока она не сломается и не перестанет поступать питающий ее электрический ток).

Дисплейная черепашка проделывает то же самое, сталкиваясь с изображенным на экране графическим аналогом механической преграды — перпендикулярной ее движению линией, квадратом или иной фигурой (предварительно на-

рисованной посредством той же черепашки или прямо рукою учащегося с помощью мыши).

Об остальных «органах чувств» и возможных формах поведения этого кибернетического механизма, похожего на организм, которым не только приятно, но и весьма поучительно командовать, мы поговорим позднее, а сейчас подведем кое-какие промежуточные итоги.

### С волнением и гордостью

Начальные занятия с черепашкой преследуют три цели:

а) «не заучивать правила, а развивать умение постигать способ, каким люди движутся в пространстве» [2, 3];

б) «обучить ребенка некоей эвристической процедуре», состоящей «в попытках установить связь между движениями собственного тела и формальными знаниями» [2, 3];

в) подвести учащегося к знакомству «с плодотворными идеями общего характера, такими, как идея иерархической организации (знаний, структур, строения организмов), идея планирования при выполнении программы, идея отладки» [2, 3], т. е. нахождения и исправления ошибок как неотъемлемой и очень важной части процесса учения.

В самом деле, «обучая» черепашку рисовать квадраты и круги по образцу своих собственных движений, ребенок сам для себя открывает, точнее — конструирует и на всю жизнь усваивает ряд фундаментальных понятий геометрии. Одновременно он учится «развивать у себя математическую стратегию: чтобы учиться чему-то, необходимо сначала осмыслить это». Иными словами, включить изучаемое новое в круг уже имеющихся представлений, найти какие-то значимые параллели и аналогии, хотя бы частично связать с чем-то знакомым и своим. Такой тип учения Пейперт называет «синтонным», т. е. «созвучным» сразу двум (или большему количеству) на первый взгляд далеких друг от друга сущностей. Так «присование черепашки круга синтонно телу», ибо оно осмысливается ребенком через связь со знанием собственного тела. А «синтонное «я» означает созвучность представлений детей о себе, как людей с определенными целями, намерениями, желаниями, симпатиями и антипатиями. Ребенок, рисующий с помощью черепашки круг, делает это с волнением и гордостью» [2, 3].

Именно синтонность такого рода за-



нятий с черепашкой делает их средством изучения множества других вещей, а также развивает способность применять полученные знания на практике. Например, усваиваемая в языке Лого математическая идея угла составляет неотъемлемую часть навигации. Однако, в то время как «многие миллионы специалистов водят суда и самолеты, читают карты» и решают другие навигационные задачи, «для большинства из них между их живой практической деятельностью и мертвой школьной математикой лежит непреодолимая пропасть». Использование же черепашки «в качестве метафорического носителя идеи угла связывает эту идею с геометрией тела», а затем и «с навигацией, деятельностью, прочно связываемой многими детьми с внешней культурой»; в результате усвоение данного геометрического понятия оказывается как «синтонным телу», так и «синтонным культуре» [2, 3].

Благодаря сочетанию перечисленных моментов «раскрытие возможностей математики становится способом жизни» и «ребенок учится рассуждать на

языке математики» так, как если бы это было для него вполне естественным повседневным делом, не омраченным никакими страхами относительно его «математической неспособности». Удается «показать, как компьютер может сформировать у нас новое отношение к математике» — самого «страшному» из всех учебных предметов, а отсюда появляются «серьезные основания заявлять, что компьютер способен изменить наше отношение к другим видам учения, которых мы можем бояться» [2, 3].

Речь идет, конечно, не о компьютере как таковом, но о формировании при его посредстве *новой образовательной культуры*, максимально благоприятствующей раскрытию всех способностей индивида к освоению любой академической дисциплины. Наименьшей «ячейкой» в этой новой культуре служит «микромир» Лого, который учащийся сперва задумывает и создает, а затем делает предметом критики и рефлексивного анализа.

(Продолжение следует)

## Литература

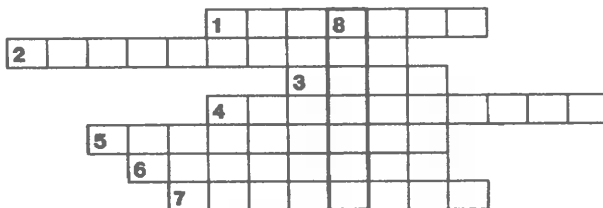
1. Сопруноу С. Ф. LogoWriter: Пособие для учителей. М.: Институт новых технологий, 1991, С. 6.
2. Papert S. Mindstorms. Children, Computers and Powerful Ideas. New York: Basic Books Inc., Publishers, 1980; 2nd ed. 1993.

3. Пейперт С. Переворот в сознании: дети, компьютеры и плодотворные идеи. М.: Педагогика, 1989.
4. Papert S. Children's Machine. Rethinking School in the Age of Computer. New York: Basic Books Inc., Publishers, 1993.

\* \* \*

## КРОССВОРД

Составитель — Г. Г. Заковряжина (г. Нягань, Тюменская обл.)



### По горизонтали:

1. Устройство для автоматической обработки информации.
2. Внешнее устройство ЭВМ для отображения графической и текстовой информации, в котором изображение строится посредством электронно-лучевой трубки.
3. Вид внешней памяти ЭВМ, устройство для хранения информации.
4. Объем информации, обрабатываемой ЭВМ за одну операцию.
5. Внешнее устройство ЭВМ, используемое для ввода информации, позволяющее пользователю вручную перемещать курсор по экрану дисплея.
6. Устройство для обработки информации в ЭВМ.
7. Упорядоченная последовательность действий для ЭВМ, реализующая алгоритм решения задачи.

### По вертикали:

8. Печатающее устройство.

**С. Ф. Сопрунов,**

*кандидат физико-математических наук, руководитель лаборатории Лого ИНТ*

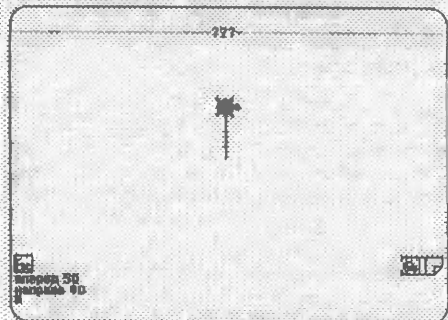
## ЛОГО-СРЕДА ДЛЯ ДЕТСКОГО ТВОРЧЕСТВА

Сегодня Лого — самая популярная в мире и, вероятно, в России среда для детского компьютерного творчества. Только в Москве Лого для IBM, Macintosh, УКНЦ и Yamaha используется в сотнях школ.

В данный номер включены статьи московских авторов, касающиеся различных аспектов использования среды Лого в образовании. Авторы этих статей — учителя, использующие Лого в своей повседневной практике.

М. А. Малышева («Дети и Лого») делится своим опытом работы с дошкольниками и младшими школьниками. Несмотря на то что это не научная статья, рассказ автора очень эмоциональный и увлекательный. Впечатления Н. С. Станченко («Фигуристы, клумбы и Чучело — образные ассоциации в ЛогоМирах») отнюдь не столь безоблачны. В ее работе содержится ряд критических замечаний, которые наверняка были бы интересны разработчикам системы. В статье рассматривается очень широкий круг вопросов — от начального этапа знакомства с Лого до программирования параллельных процессов, что зачастую представляет трудности и для достаточно опытных программистов. О. Г. Кутукова («Методика поведения итогов») обсуждает в основном вопросы методики и опыт коллективной работы учеников над проектными заданиями. Эти актуальные методические проблемы рассматриваются в достаточно общей постановке — Лого является лишь одним, хотя и весьма важным, конкретным примером. Статья А. Г. Юдиной на первый взгляд столь же неприязнительна, как и ее название — «История с зайцем». В ней рассказывается всего лишь о первых шагах проекта, над которым работают два девятиклассника. Однако автору удалось убедительно продемонстрировать, как личная заинтересованность учеников в реализации собственного замысла заставляет ребят взглянуть по-новому на традиционно трудные (и скучные) школьные дисциплины, например тригонометрию. И наконец, А. Л. Горбунов («Компьютерные сети для детей через LEGO TC Logo») описывает конкретное моделирование компьютерных сетей в среде Lego TC Logo.

Работа, выполняемая в Лого, представляет собой нечто вроде «компьютерного альбома», состоящего из одной или нескольких «страниц», которые можно листать на экране. На страницах «живут» черепашки. Они являются основными (хотя и не единственными) объектами, которыми управляют с помощью языка Лого. Управлять черепашками можно непосредственно (интерактивно) для чего в специальной области экрана — в «поле команд» нужно написать соответствующие команды, которые будут немедленно выполнены.



Результат выполнения двух команд — вперед 90 и направо 90. Первая команда перемещает черепашку на заданное расстояние, вторая — поворачивает черепашку направо на 90 градусов.

Перед выполнением команд черепашка находилась в центре страницы, ориентирована головой вверх, первоначально.

Нет смысла, даже вкратце, объяснить все команды Лого — обычно их более 200. Среди них есть и графические команды: команды перемещения черепашки, изменения цвета, изменения формы (черепашка может, например «надеть на себя» форму самолета или человечка) и т. д. Есть команды, позволяющие создавать музыку, выполнять арифметические вычисления и прочее.

Удивляет разнообразие представленных статей. Что общего между созданием кукольного спектакля «Красная Шапочка» (О. Г. Кутукова), моделированием трехмерной графики (А. Г. Юдина) и протоколом DATA LINK PROTOCOL, входящего в спецификацию OSI международной организации стандартов ISO (А. Л. Горбунов)? Что объединяет первоклассников — «прирожденных программистов» (М. А. Малышева) и девятиклассников, на которых преподаватель «махнул рукой (как на юных программистов)» (А. Г. Юдина)? Стоит ли вообще искать что-то общее в тематике заданий и стиле преподавания столь разных учителей?

На первый взгляд ответ на этот вопрос давно известен. Разработчики Лого неоднократно заявляли, что Лого «всего лишь инструмент», подобный карандашам и альбому с красками, что Лого не навязывает учителю, как и для чего его использовать. Цель разработчиков — предоставить учителю и ученикам разнообразные и удобные средства реализации собственных замыслов. В таком ответе, несмотря на его правдивость, есть некоторое лукавство. Каждый знает, что мастер выбирает из множества инструментов тот, который «ему по руке» (конечно, если есть выбор, но в наше время большинству учителей есть из чего выбирать), а используемые инструменты неизбежно влияют на то, как и для чего мы их используем.

Я не берусь ответить на все поставленные вопросы. Есть ли что-то, что объединяет большую часть учителей, выбравших для себя Лого и работающих с ним? Часто приходится слышать о «Лого-сообществе» (национальном или интернациональном). Что означает этот термин? Есть ли разница между терминами «Лого-сообщество» и, скажем, «Бейсик-сообщество»?

Я приглашаю вас познакомиться с впечатлениями пяти учителей, работающих с Лого. Для читателей, впервые встречающихся с этим языком программирования, мы приводим краткие предварительные сведения о системе. Также во все статьи, кроме статьи А. Л. Горбунова, в которой представлены достаточно сложные для первоначального знакомства с Лого тексты программ, включены комментарии для новичков. Неважно, работали вы уже с Лого или встречаетесь с этим инструментом впервые, — надеюсь, что в любом случае эти статьи будут вам интересны.

Кроме непосредственного управления в Лого доступен и режим программирования. В особом текстовом окне на экране компьютера вы можете записать программы, которые, как и в большинстве языков программирования, «расширяют исходный словарь языка» — имена записанных вами программ можно использовать как в интерактивном режиме, так и при записи новых программ.

Запись программы всегда начинается с заголовка (строки, содержащей слово это и имя программы) и завершается словом конец.

Вот пример простой программы, рисующей квадрат:

```
это квадрат
  повтори 4 [вперед 60 направо 90]
конец
```

(команда повтори 4 [вперед 60 направо 90] заставляет черепашку 4 раза повторить команды, заключенные в квадратные скобки).

В настоящее время самой популярной в России версией Лого для компьютеров IBM является система LogoWriter для компьютеров Macintosh — система Лого-Миры, обладающая, по сравнению с LogoWriter, существенно более богатыми возможностями. Обе системы в некотором смысле двуязычны — можно использовать как русские, так и английские имена команд.

Несколько особняком стоят системы Lego TC Logo (компьютеры IBM и Apple) и ControlLab (компьютеры Macintosh и IBM). В этих системах, используя язык Лого, можно управлять механизмами и роботами, собранными из специальных конструкторов Lego, содержащими, кроме обычных деталей Lego, различные моторы, датчики и т. д.

**О. Г. Кутукова,**

*средняя школа № 161, Москва*

## МЕТОДИКА ПОДВЕДЕНИЯ ИТОГОВ

Известно, что ученикам нравится получать хорошие оценки, и не нравится, когда им ставят двойки. «Синдром оценки» у детей начинает развиваться еще в дошкольном возрасте с помощью родителей и воспитателей детского сада, закрепляясь в школе. Не менее известно, что обществу «отличники-зубрилки» так же мало нужны, как «двоечники», потому что ни те, ни другие не умеют применять свои знания. У одних они благополучно забыты после проверки, у других и вовсе не получены.

В последнее время жизнь показывает, что преуспевает только тот, кто умеет активно применять свои знания и опыт. Как правило, эти люди учились и совершенствовались по собственной инициативе, и добытые ими знания оказываются более долговечными.

Учиться таким активным и творческим образом помогает использование зачетной системы оценки знаний. Завуалированная оценка от 3 до 5 в слове «зачет» позволяет меньше затрагивать самолюбие ребенка, зато знания, полученные в ходе работы над проблемой, «добываются» им самим, и поэтому останутся надолго и вспомнятся в нужный момент. Для некоторых предметов (например, литература, окружающий мир, информатика) такой вид деятельности позволяет еще лучше выявить способности каждого, развить их и усовершенствовать почти индивидуально.

Целесообразно при этом использовать дружеские связи для активной работы после уроков, организовать учащихся по группам. Как правило, дети сами легко распределяют обязанности по уже сложившимся интересам. Задача учителя — следить только за тем, чтобы деятельность каждого члена группы была разнообразна.

При этом учитель может столкнуться и с желанием некоторых учеников работать только самостоятельно. Это может происходить либо в силу сложившейся в классе обстановки, либо в силу «честолюбивой жадности и ревности» самого учащегося и его нежелания делиться своим успехом с другими. Практика показывает, что по ходу работы многие индивидуалисты начинают искать пути сближения с товарищами по работе или «несчастью», чтобы разделить бремя большой подготовительной работы. Но бывает, что необходимая работа проводится учеником

самостоятельно и хорошо. В этом случае, безусловно, его знания будут крепче, чем у ребят, работавших в группе. Если же это индивидуалист-лентяй, то учителю для таких ребят необходимо устраивать отдельный контроль знаний только по материалам учебника.

Наиболее оптимальное число участников в группе для выполнения большого тематического задания, выданного на целую четверть или полугодие, — 4—5 человек, имеющих одинаковый уровень знаний. Для групп с разным уровнем подготовки целесообразно иметь ответственного за отдельные куски работы или фамилию «автора» созданной части. Благодаря этому гораздо легче проследить за ростом творческого и «научного» потенциала каждого ребенка и ввести индивидуальную корректировку оценки, если это необходимо. Как правило, в таких группах реальным лидером становится наиболее знающий и работоспособный ученик, и частое общение с ним менее активных и подготовленных детей тоже имеет свои положительные стороны. Над заданием, рассчитанным на один урок, удобно работать по парам. Ученикам разрешается пользоваться литературой, помогающей найти правильный ответ, причем даже на контрольных. Наличие такой официальной подсказки очень облегчает психологическое состояние ребенка, снижает его неуверенность. Для нас же ясно, что ученик, изучавший материал, быстро найдет и прочитает правильный ответ, а не выдумает свою неправильную версию.

Однако в этой методике есть два сложных момента. Во-первых, шум на уроке. Но это качественно другой шум: увлеченные работой ученики бурно выражают свои эмоции, не обращая внимание при этом на шум других, которые часто мешают проводить продуктивные занятия в классе по стандартной схеме. Во-вторых, зачетная методика «научить обучая» требует постоянного совершенствования знаний самого учителя и серьезной проработки урока. От того, как вы подготовите работу каждой группы, зависит ее активность.

Для проведения такого творческого урока (лучше, чтобы он был двоянным) большое значение имеет обеспечение рабочей обстановки с необходимыми для нее атрибутами и соответствующий возраст учеников.

План урока:

1. Рассадить детей по группам (сдвинуть мебель, обеспечить «неприкосновенность территории» каждой группы).

2. Раздать задание, литературу, необходимый подборный материал, для лабораторных работ заранее приготовить оборудование.

3. Объяснить главную задачу и требования к ее исполнению.

4. Следить за ходом выполнения работы и помогать по мере необходимости.

5. Проверить качество выполнения этапа во время урока или собрать материал для домашней проверки.

6. Напомнить домашнее задание.

7. Привести класс в порядок.

Часть понятия «обеспечение рабочей обстановки» уже отражено в плане урока. Дополнить необходимо следующее:

1. Учитель заранее продумывает форму отчетности по данному вопросу.

2. На предварительном уроке обязательно рассмотреть варианты оформления и частично их подготовить. Желательно иметь уже идеально выполненный вариант для демонстрации.

3. Собрать (подготовить самому или с помощью учеников) вопросы по исследуемой теме, ограничить круг наиболее важных вопросов, решению которых будет посвящено следующее занятие.

4. В задании, написанном для каждой группы, выделить общие важные вопросы и дополнить, с учетом интересов и возможностей ребят.

5. Для домашней подготовки дать список необходимой литературы и подборного материала.

Приведу несколько примеров конкретного воплощения этих рекомендаций в применении к урокам информатики.

1. Урок информатики является подержкой другого урока.

1.1. Литература, история, иностранный язык.

Такой вид деятельности предполагает либо тесный контакт с учителями-предметниками, либо высокий уровень эрудиции самого учителя информатики. Целесообразность таких уроков информатики в классах гуманитарной направленности очевидна (в нашей школе таких уроков не было). В этом случае первая четверть посвящена освоению текстового редактора, совмещенному с выполнением конкретных упражнений для беглости пальцев и повышения грамотности. В качестве зачета можно использовать:

а) диктант для парной работы: наборщик—корректор;

б) набор и редактирование статьи в школьную газету (одному или двум ученикам);

в) составление, набор, редактирование и распечатка серии заданий для

младших классов, на отдельные темы (для группы учеников).

При выполнении заданий оценивается умение использовать все изученные возможности редактора, грамотность исполнителей, оформление, значимость заданий.

В следующих четвертях добавляется изучение графического редактора и на зачет можно вынести большие тематические работы:

а) создание мини-книжек с картинками по интересующим темам. Если над таким заданием работает группа, то в ней может произойти разделение труда: кто-то будет подбирать материал из учебника или другой литературы, кто-то займется созданием иллюстраций; часть детей поочередно будет набирать и редактировать текст; возможно дружеское рецензирование и, наконец, окончательное оформление с использованием цвета;

б) выпуск толкового словаря с иллюстрациями двух типов: большая картина с использованием лексики и номерами переведенных и объясненных слов или серия алфавитных картинок с переводом и примерами применения;

в) изготовление карточек-картинок в помощь учителям младшей школы. Польза от такого взаимного общения неоценима;

г) подготовка иллюстрированного каталога основных исторических событий;

д) написание и художественное оформление реферата по теме, полученной на другом уроке, в том числе и по информатике;

е) школьники более старшего возраста могут составить серию тематических баз данных с любым набором интересных объектов;

ж) составление инструкции проведения лабораторной работы по любой теме (желательно с учетом реального воплощения в жизнь на соответствующем уроке. Главное условие таких инструкций — их оригинальность и новизна. Однако на худой конец можно «усовершенствовать», а главное, лучше проиллюстрировать объяснение опытов, регулярно проводимых по предмету);

з) решение задач по предметам с учетом знаний информатики.

1.2. Уроки естественно-научного цикла.

Для них применимы все вышеперечисленные виды зачетов, так как изучение материала с иллюстрациями более интересно.

1.3. Тематическое моделирование в рамках определенной программы.

В последнее время созданы и рекомендуются к использованию большие многоцелевые программы, в которых ученик является режиссером действия

главных героев. Однако без знания основных тем по предмету (или нескольким предметам) трудно успешно «прожить в программе». Так, например, в самых распространенных и «простых» сказках-лабиринтах вы обязаны провести героя к обозначенной цели, но без знания английского языка, умения анализировать ситуацию на экране, логического рассуждения и специальных знаний по определенной теме (правила грамматики, математики, физических законов, юриспруденции, биологии и т. д.) ваш герой далеко не уйдет.

В качестве зачета в данном случае можно вынести:

- а) перевод основных диалогов;
- б) составление «карты» маршрута по лабиринту;
- в) составление перечня использованных знаний по предметам;
- г) составление инструкции прохождения того или иного участка пути.

В программах другого типа, например, «Моделирование и конструирование» просто необходим тесный контакт с учителем-предметником для грамотного составления зачетных заданий. Приведу примеры только очевидных заданий:

- а) сконструировать «внешний вид» молекул определенного вещества из его химических элементов;
- б) сконструировать лабиринт для прохождения шарообразного тела, используя законы физики;
- в) исследовать влияние изменения состава атмосферы на жизнь определенного природного сообщества;
- г) моделирование процесса мутации одного из видов живых организмов;
- д) исследование влияния действия вулканов на все сферы Земли;
- е) моделирование жизни города с разными начальными условиями;
- ж) создание экологически чистого крупного поселения людей;

2. Урок информатики — часть физико-математического образования.

В таком виде уроки информатики существуют с момента своего появления, и программа таких уроков постоянно совершенствуется. Темы зачетов зависят здесь не только от изучаемого языка программирования, но и от уровня знаний учеников. На зачет, как правило, выносятся создание самостоятельной программы на определенную тему или программа решения конкретной задачи. В общеобразовательных классах целесообразно использовать для этого групповую форму зачета с разделением труда, однако основные способы употребления тех или иных команд и типов программ должны быть проверены непосредственно у каждого ученика отдельно. Обязательным условием защиты программ такого

типа является представление ее наиболее слабо подготовленным членом группы, который, может быть, даже и не принимал участия в ее составлении. Кроме того, любая программа должна быть хорошо оформлена и иметь инструкцию.

Темы таких зачетов неоднократно приводились на страницах журнала «Информатика и образование» и часто используются учителями. Однако хочется напомнить, что не менее важны программы контроля знаний как по информатике и конкретному языку программирования, так и по темам всех других уроков для любых классов. В качестве примера приведу программу под названием «Суффиксный футбол», пользующуюся неизменным успехом у учащихся III—V классов. В ней смешной человек предлагает детям познакомиться и поиграть в мяч. На мячике написаны суффиксы «н» или «nn», и ученик должен направить его к нужному слову, которое для него припас Словознайкин. Программа создана одиннадцатиклассником г. Фурманова, а усовершенствована и дополнена в нашей школе.

В младших классах целесообразно начинать с обучения встроению программированию поведения отдельных исполнителей или при работе в средах мультимедиа.

Принцип «научись сам — научи другого» прекрасно реализован в среде Лого и, особенно, в его усовершенствованном варианте LogoWriter. В качестве зачета здесь можно использовать создание мультфильма на любую тему. Вот названия некоторых из них:

Кукольный спектакль «Красная Шапочка».

«Случай по дороге в школу».

«Несколько сюжетов из жизни морского дна».

«Рождение Афродиты».

«Все подвиги Геракла».

«О пользе уроков физкультуры».

«Открытие законов физики».

«Гонки «Париж—Дакар».

«Последний подвиг» (о героях Великой Отечественной войны).

Это далеко не полный список созданных мультфильмов, идеи которых рождаются у самих детей. Задача учителя — постоянно совершенствовать методы их воплощения, постепенно изучая новые команды и алгоритмы построения программ.

В заключение следует еще раз подчеркнуть, что система зачетной оценки знаний является одним из наиболее эффективных методов изучения материала по любому предмету как в старших, так и в младших классах. Она позволяет максимально мобилизовать учителя и ученика для совместной творческой работы.

Н. С. Станченко,

учитель информатики средней школы № 1747, Москва

## ФИГУРИСТЫ, КЛУМБЫ И ЧУЧЕЛО — ОБРАЗНЫЕ АССОЦИАЦИИ В ЛОГОМИРАХ

### Растительная пища

В фигурном катании есть понятие «школа». Это вычерчивание коньками на льду определенных фигур в различных направлениях. В «школе» оценивается только техника исполнения. Красивые костюмы, подобранная музыка, хореография, театральные жесты — все, что нужно для создания образа, для зрелищности, мы видим только на выступлениях перед зрителями. А фигурист в спортивном костюме, выполняющий упражнения, вызывает меньше интереса.

Всем известная черепашка из ЛогоМиров оставляет тонкий черный след на белом фоне листа, как конек фигуриста на льду. Но можно изменить цвет листа, задать узор и цветное перо разной толщины. После этого треугольник, или круг, или квадрат, нарисованные черепашкой, выглядят более привлекательно. В ЛогоМирах есть и другие возможности разнообразить картинку, а заодно и занятия. Для этого черепашка, подобно фигуристу, должна войти в «образ», т. е. «надеть новую форму». Для таких превращений понадобятся команды **нов\_форма**, **штамп**. Теперь использование команды — **повтори** в задачах по геометрии — приобретает новый смысл.

\*Команда **нов\_форма** изменяет форму черепашки («одевает на нее новую форму»). В качестве параметра команде сообщается номер или имя нужной формы. В Лого кавычки ставятся только в начале слова. В системе ЛогоМиры имеется большое количество уже готовых форм хотя, конечно, пользователь может рисовать новые и менять уже имеющиеся формы.

Команда **штамп** «штампует» форму черепашки в том месте, где стоит черепашка.

Команда **повтори** представляет собой простейший цикл — она повторяет указанное число — последовательность команд, заключенную в квадратные скобки.

Ребенок может почувствовать себя и строителем, и садовником, и командиром флотилии.

Посадим деревья. Черепашка в исходном положении:

```
нов_форма "дерево
повтори 10 [ штамп вперед 15 ]
```

С севера на юг вырастает аллея де-

ревьев. Можно посадить салат на грядке. Получилось? Теперь разобьем клумбу. После этого задания легко усваивается движение по окружности. А многоугольники? Это дома на квадратной площади, эскадрилья самолетов (равносторонний треугольник), созвездие (произвольный многоугольник)... Примеров здесь множество: ели на горке (полуокружность), стоянка машин (узлы решеток), венки сердечек, стая птиц, демонстрация, табун лошадей...

Все эти картинки появляются на экране при задании движения черепашке по какой-нибудь геометрической фигуре или кривой и использовании команды **штамп**.

Предварительно можно «подготовить почву». Например, сначала черепашка рисует газон:

```
нов_цвет "зеленый
нов_узор 17
крась
```

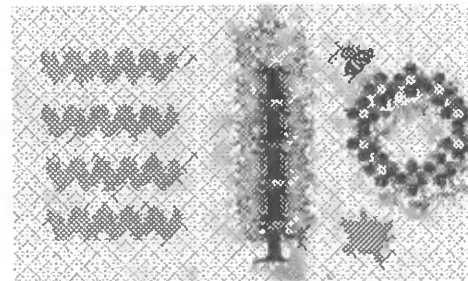
Команда **нов\_цвет** изменяет цвет рисующего пера черепашки. Команда **крась** заливает краской область на экране, внутри которой находится черепашка. Цвет заливки определяется цветом пера черепашки. Заливка не обязана быть сплошной — можно указать узор заливки с помощью команды **нов\_узор**.

Многие команды в Лого имеют сокращенную форму, **вп** — сокращение для команды **вперед**, **пр** — сокращение для команды **направо**.

Затем превращается в цветы на клумбе:

```
нов_форма "цветок
повтори 10 [ штамп вп 30 пр 36 ]
```

Пусть теперь над цветами летает пчела:



нов\_форма "пчела  
повтори 30 [ в 15 пр 12 ]

Таким образом, для создания простейшего мультфильма «Лето» понадобилось совсем немного команд, а использование сведений из геометрии спрятано в красивых формах цветов и летающей над ними пчелы.

Если вдруг появится черепашка *ич* "черепаха (*ич* — сокращенная форма команды *нов\_черепашка*, создающей новую черепашку с указанным именем), то она вполне сможет сказать стихами Эмиля Виктора Рью\*:

*Монолог черепахи, вновь посетившей  
через полчаса грядку с салатом,  
хотя ей уже давно было пора  
вкушать послеобеденный сон  
на клумбе среди голубых незабудок*

*Растительная пища —  
Такая вкусотища!*

(Перевод Г. Кружкова)

### Ужасная песенка

*Чучело-Мяучело  
На трубе сидело.  
Чучело-Мяучело  
Песенку запело...*

Так распевали две веселые маленькие девочки, рисуя в KID PIX свою картинку. Продолжение этого стихотворения я вспомнила на другом занятии, когда мы захотели послушать мелодию, записанную в музыкальном редакторе ЛогоМиров:

*Чучело-Мяучело  
С пастью красной-красной,  
Всех оно замучило  
Песенкой ужасной...*

Песенка действительно оказалась всем ужасной после нескольких попыток ввести ее имя из командной строки. В ответ вместо музыки мы получали сообщение об ошибке. Попытались назвать песенку по-другому, но, перепробовав

безрезультатно несколько имен, заметили, что все они выстроились в длинный список в меню мелодий. Это было удивительно, ведь нотная запись песенки не менялась. Значит, переименование не происходило, а каждый раз эта ужасная песенка записывалась под очередным именем! Но ни под каким из них она не хотела исполняться в нашем проекте. Разобраться в ошибках нам помогла сотрудница ИНТО Ирина Бахметьева: у всех имен было общее свойство — они начинались с пробела...

В ЛогоМирах мы даем имена мелодиям, звукам, листам, черепашкам, бегункам, формам... И правильно набитое имя — уже полдела. Но бывает, что проект и не начинает выполняться, так как при первом же обращении по имени выдается обескураживающее сообщение, что такого имени нет. Ошибки из-за лишних пробелов происходят довольно часто. Если пробел встречается в середине имени, то он сразу заметен, и такая ошибка быстро исправляется. Но пробелы в начале и конце имени видны только в диалоговом окошке. Самые «вредные» из них появляются незаметно. В диалоговом окне, где дается имя, курсор прижат к стенке, и многие дети непроизвольно делают «красную строку», тем самым набивая пробелы, а затем имя\*\*. Казалось бы, пробелы в начале имени должны игнорироваться, но это не так. И «невидимки» в именах заставляют новичков опускать руки. Ведь пока такое имя не используется в инструкции, все проходит незаметно и лишь при обращении к нему появляется сообщение об ошибке, работа останавливается. В конце имени пробелы набираются реже, ошибки чаще происходят в процедурах, когда путают клавиши «ввод» и «пробел» после набора имени процедуры в строке «это...».

Но вернемся к нашей песенке. Стихотворение Михаила Яснова\*\*\* «Чучело-Мяучело» бесконечное. Таким же бесконечным мог бы быть список имен одной и той же мелодии, если бы мы продолжали ее переименовывать. Кстати, команды `список_мелодий` нет — все имена

\* Эмиль Виктор Рью (1887—1972) — английский ученый и переводчик древней поэзии. Чтобы отдохнуть от серьезных занятий, писал для детей смешные стихи. Часто их названия оказывались гораздо длиннее самих стихотворений. Видимо, сказывалась привычка давать подробные названия своим научным трактатам.

\*\* Эта ошибка встречается настолько часто, что ее, наверное, можно отнести к проблемам психологии

\*\*\* Михаил Яснов (род. 1946) — поэт и переводчик.



можно просмотреть только в меню «Мелочи». Там может встретиться пустая строка, которая «работает» как имя, хотя есть сообщение: «имя не может быть пусто». При большом количестве мелодий в проекте компьютер сообщает об отсутствии свободной памяти. Можно попытаться удалить лишние мелодии, однако просто «вырезать» или «отменить» мелодию нельзя. Избавиться от песенки можно лишь с помощью команды «удали». При этом, как в случае с многословным именем, придется огородить песенку со всеми ее пробелами высоким забором: удали " | песенка |.

Команда **удали**, как и следует из ее названия, удаляет объект с указанным именем.

## Реализоваться в собственном мире

Изучение черепашкой графики на наших занятиях оживляло использование различных возможностей ЛогоМиры. Мы делали черепашку цветной, изменяли размер ее пера, использовали отпечатки ее форм и разные расцветки листа. Но мы быстро поняли, что, изменив форму черепашки на другую, получим не новую управляемую фигурку, а подвижную картинку, не меняющую, как черепашка, свою ориентацию.

Знакомство с мультипликацией позволило «оживить» формы черепашки. Экран заполнился летающими птицами, пешеходами, скачущими лошадьми, мчащимися автомобилями...

Мы стали думать о содержании мультфильма, записывать к нему звук и мелодии. Наши команды черепашкам все больше подчинялись сюжету...

События в мультфильмах развиваются по-разному: и одновременно, и друг за другом, и совершенно независимо. Чтобы управлять этой «жизнью», мы занялись изучением параллельных процессов, а потом и их синхронизацией.

Для объяснения механизма управления процессами детям предлагалось сделать мультипликацию на разные темы.

Использование множества примеров взаимодействия процессов позволяло каждому ребенку решать индивидуальную задачу, что повышало творческий интерес и давало возможность учесть уровень подготовки и различные при-

страстия учеников. Например, мальчики с удовольствием делали мультфильмы о технике, а девочки — о животных, растениях... Таким образом, сложные проблемы программирования, наполненные художественным смыслом, становились незаметными, и дети легче с ними справлялись. Конечно, при создании мультфильма требуется много времени на красочное оформление, которое кому-то покажется тратой времени на уроке, но это не так. Во-первых, использование графического редактора и редактора форм разнообразит занятие и дает время для «отдыха». Во-вторых, для успешного завершения работы требуется умение ее организовать поэтапно. Обычно это происходит в такой последовательности: создание образов, форм черепашек, среды их обитания и собственно программирование. В-третьих, удовлетворяются интересы детей как с техническими, так и с гуманитарными наклонностями, так как любой этап работы может стать «самым важным» и наиболее разработанным.

Таким образом, программа ЛогоМиры благодаря своей многогранности дает возможность каждому ребенку реализоваться в собственном мире.

Рассмотрим задачи с мультипликацией, в которых требуется добиться синхронизации двух процессов. Первый из них запущен командой **всегда** и продолжается бесконечно. Другой — выполняется по команде **запусти** и через некоторое время заканчивается. Одновременно с этим останавливается и первый процесс\*.

В системе ЛогоМиры процессы могут идти параллельно. Например, на экране могут одновременно двигаться несколько черепашек, управляемых независимыми процессами. Вот несколько команд, позволяющих контролировать параллельные процессы:

Команда **всегда** запускает новый процесс, представляющий собой бесконечное повторение указанных команд. Например, выполнение команды **всегда [вперед 1 направо 1]** приведет к движению черепашки по кругу.

Команда **запусти** запускает новый процесс, однако указанные в квадратных скобках команды выполняются однократно. Например, выполнение команды **запусти [повтори 1000 [вперед 1 направо 1]]** приведет к тому, что черепашка пройдет 1000 шагов по кругу, и после этого процесс завершится.

Команда **готово?** сообщает, завершился ли уже указанный процесс. Команда **жди\_пока** приостанавливает выполнение программы до тех пор, пока указанное в квадратных скобках условие не будет выполнено: например, команда **жди\_пока [готово? [процесс1]]** приостановит

\* Схема синхронизации содержится в пособии ИНТ «ЛогоМиры».

работу до тех пор, пока не завершится «процесс1».

Команда `жди` вызывает паузу при выполнении процесса, время паузы определяется значением параметра.

Для создания этой модели нам понадобятся две черепашки и знание нескольких команд. Команда `жди_пока` поможет синхронизировать наши процессы, а датчик `готово?` сообщит об окончании процесса, начавшегося по команде `запусти`.

Чтобы остановить бесконечный процесс, используем команду `отмени`. Часы с маятником — пример взаимосвязанных процессов. Одна черепашка принимает форму циферблата с различным положением стрелок, а другая — форму маятника в равновесии и с наклонами вправо и влево.

```
это процесс1
нов_форма "стрелки1 жди 5
нов_форма "стрелки2 жди 5
нов_форма "стрелки3 жди 5 конец
```

```
это процесс2
повтори 30[нов_форма "маятник1 жди 3
нов_форма "маятник2 жди 3
нов_форма "маятник3 жди 3
нов_форма "маятник2 жди 3] конец
```

```
это часы
черепашка1, всегда [процесс1]
черепашка2, запусти [процесс2]
жди_пока [готово? [процесс2]]
отмени [процесс1] конец
```

Стрелки часов могли бы двигаться бесконечно, но, когда прекращает раскачиваться маятник, они останавливаются. Программы имеют возможность дополняться боем часов, появлением «кукушки» и другими подробностями.

Эта же схема используется при создании мультфильма «Надоедливая оса».

Оса летает над лицом по сложной траектории, глаза за ней «следят» (смена форм лица с разным расположением зрачков). Когда оса останавливается, лицо тоже «замирает». Здесь возможно такое развитие сюжета: оса улетает совсем, а лицо исчезает, т. е. начинаются новые процессы. Лицо можно заменить на фигуру человека, размахивающего руками, добавить звук жужжания, крики...\*



Вызывает большой интерес и задача о поезде, где надо синхронизировать работу электровоза и вагончиков. Электровоз начинает двигаться по команде `запусти`, а вагончики-черепашки — по команде `всегда, каждая`. Когда останавливается электровоз, встает и весь поезд. Понятно, сколько дополнительных деталей можно придумать еще в этой ситуации.

Примеры таких задач можно продолжить. Важно, что многие из них придуманы детьми. Пофантазируйте сами, может быть, и у вас появятся интересные мультфильмы с новыми фантастическими сюжетами, а исполнить ваши идеи помогут черепашки из ЛогоМиров.

\* Рисунки выполнены учащимся школы № 1747 Москвы Сергеем Олейником.

**М. А. Малышева,**

*преподаватель информатики, г. Обнинск*

## ДЕТИ И ЛОГО

С февраля 1994 г. я начала вести кружок «Информатика для малышей» в Обнинском ЦНТГУ «Эврика», где уже несколько лет действует студия «Развивающие игры». В традициях студии — проведение совместных занятий детей и родителей. Это означает, что родители не только присутствуют на занятиях, но и активно в них участвуют: помогают понять задание, поощряют, иногда подсказывают путь к решению. Занятия часто строятся в форме сказки, сюжет которой иногда развивается в течение нескольких дней.

Многое из опыта студии использовалось и в работе кружка «Информатика для малышей», целью которого было введение в информатику, развитие и подготовка к школе детей 5—6 лет. В занятиях принимали участие и школьники 7 и 9 лет, но они занимались, как правило, без родителей, поэтому им приходилось гораздо труднее. Опыт показал, что при работе с компьютерами детей младшего возраста участие взрослых особенно важно, даже если они тоже видят компьютер впервые.

Во время работы кружка дети участвовали в развитии сюжета сказки про путешествие мальчика Сережи по волшебной стране Инфории. Они помогали Сереже и его друзьям справляться с кознями злого волшебника Форбеяса. В конце концов злодей был изгнан из Инфории, а вездеходик МИК (Маленький Исполнитель Команд) осуществил свою мечту и превратился в черепашку. Далее мы уже учились писать программы для черепашки на языке Лого.

По нашему мнению, Лого важен даже не как язык программирования, а как средство развития личности, познания мира. Ребенок учится анализировать любую проблему, учится относиться к любой ошибке не как к катастрофе, а как к тому, что следует найти и исправить. Лого позволяет проявить себя в творческой деятельности детям с математическим складом ума, которые, как правило, лучше учатся в школе, но вынуждены в основном действовать как исполнители:

решать задачи, поставленные перед ними взрослыми.

Каждое наше занятие начиналось с разминки. Сначала в нее входили игры на внимание, координацию, логические и шуточные задачи. Позже, когда по ходу сказки дети знакомились с новыми персонажами, появлялись и новые игры, уже характерные только для данного кружка.

Робик нужно было помочь быстро решить примеры в соответствии с заданной в него «программой» — карточкой с надписью «+1»; «-2»; позже — «+3-1» и т. д. Дети шестилетнего возраста считают по-разному, и здесь удобно было варьировать задания: одному предложить «2+3», а другому — «38+3». Если программа становилась «секретной» (то есть в кармашек картонного Робика карточка с «программой» вставлялась надписью внутрь), задание усложнялось: нужно было найти алгоритм по входным (дети называли числа) и выходным (Робик говорил ответ) данным (аналог — Буквоед в «Роботландии»).

Чтобы накормить Инфорика (а питается он информацией, причем любит самую свежую), ребятам приходилось задавать друг другу задачи и тут же решать их. Причем разделение задачи на условие и вопрос, что часто вызывает затруднения у первоклассников, они усвоили играючи. Задачи дети придумывали тоже в соответствии с личными интересами и умением считать. А Инфорик, мохнатое зеленоглазое существо-варежка, был очень доволен и благодарил как автора задачи, так и «кормильца», первым ее решившего.

По сюжету сказки на каждом занятии приходилось преодолевать препятствия и решать задачи как с помощью компьютера, так и без него. Например, разложить по множествам листья с двух деревьев — «красного» и «треугольного», чтобы найти МИКа, превращенного в лист; подсчитать количество букв А в надписи, а затем под камнем с такой же цифрой найти в лабиринте горючее; скопировать с доски в тетрадь «забор» сложной формы, чтобы он исчез с нашей

дороги; собрать мост, разломанный на кусочки; написать для МИКа алгоритм прохода через лабиринт (все дети, занимающиеся в кружке, знали буквы, большинство могли читать); помочь Робику вырастить антенны, чтобы он связался с центральным компьютером и узнал нужную нам информацию, и т. д.

Поскольку работали мы с сетью УКНЦ, где был вначале только Бейсик, все программы приходилось создавать «на ходу», к каждому занятию. Мы попадали в Инфорию на РАКЕТЕ, которую нужно было заправить двумя сортами топлива, направляя летящие сверху кусочки в правый и левый баки (кусочками могло быть все, что можно сортировать: группы по 4 и 5 звездочек, гласные и согласные буквы, арифметические примеры с ответами 13 и 14, английские слова со звуками [i] и [ai]). Позже мы занимались сборкой МОСТА из осколков разной формы, на которые он развалился: из прямоугольников-доминошек, из кусочков с цифрами (в прямом и обратном порядке), из кусочков со словами (каждое следующее слово должно было начинаться с последней буквы предыдущего). Играли с компьютерным Робику — аналогом картонного, с той лишь разницей, что для компьютерного варианта главным была скорость решения, а также выращивали деревья, быстро подсчитывая количество «семян», мелькнувших на экране, или, печатая ненадолго появившееся слово (зрительный диктант), помогали обезьянкам добраться до кокосов (аналог — Ханойские башни), причем все дети справились с решением для трех обезьян без подсказки родителей, а многие требовали четырех. На последующих занятиях ребята проходили с МИКом все более сложные лабиринты, обучая его новым командам.

Система команд для МИКа создавалась по подобию Лого, только в очень упрощенном варианте: В — вперед на одну клетку, П — поворот направо на 90 градусов (а не шаг вправо!), Р — рисуй... По мере увеличения количества клеток в лабиринтах задания усложнялись. От управления в диалоговом режиме (по одной команде) мы старались перейти к редактированию алгоритма в целом (до 25 команд). А иначе невозможно было бы пройти через подводный лабиринт, в котором нельзя останавливаться! К тому же всем хотелось посмотреть «мультик» и полюбозаться, как МИК выполняет все, чему его научили!

Как и положено исполнителю, МИК говорил «НЕ ПОНИМАЮ», если встречал незнакомую команду, и «НЕ МОГУ», если его направляли на стенку лабиринта. При этом место ошибки как в лабиринте, так и в программе выделялось другим цветом, что очень облегчало детям редактирование. Помогали им и модели МИКа из спичечных коробков с пометками П и Л — (ведь очень трудно понять, куда поворачивать, если стоишь вниз головой!) Впрочем, часто много было наблюдать юного программиста, пытающегося вывернуть шею, чтобы заглянуть в экран сверху.

И вот, наконец, наш МИК стал черепашкой, у нас появилась дискета с Лого. Для начала пришлось написать для черепашки на английском варианте ЛОГО все процедуры МИКа, чтобы дети осваивали новую для них среду, используя уже знакомые им команды. Потом мы обнаружили, что не слишком удобно писать «В В В В В», и решили, что неплохо бы черепашке научиться выполнять команды с параметром. Переход к обычной системе команд Лого прошел на удивление легко, и дети начали писать программы. Одним это понравилось больше, чем игры, другим — не очень. Но игры тоже не закончились, появились даже новые, но уже на Лого. Так мы освоили координаты (если писать для «пчелки» черепашки координаты цветов, она перелетает с одного цветка на другой, оставляя за собой след в виде изображения рыбки, корабля или самолета). Позднее мы занялись сложением двузначных чисел, возводя башенки из кубиков, и составлением слов из «рассыпавшихся» букв. Иногда возвращались и к прежним играм. За время, оставшееся до летних каникул, мы изучили не так много «черепаших» команд, но несложные рисунки ребята уже могли создавать сами.

Немного позже мы купили программируемую игрушку «Планетоход», у которой система команд очень похожа на Лого, но повороту на прямой угол соответствует не 90, а 21. Как ни странно, детей это ничуть не смутило, и они часто вместо разминки гоняли его по всему классу, с особенным восторгом воспринимая его гудки и запуск пропеллера.

В прошедшем 1994/1995 учебном году я работала в саду-школе «Муравушка» в классе, оборудованном машинами IBM PC/386. На уроках мы пользовались обширным парком развивающих игр для IBM и работали в тетрадах экономи-

ко-статистического института (к сожалению, рассчитанных на безмашинный вариант). На кружковых занятиях можно было изучать редактор LogoWriter. Желаящими первые полгода были практически все. Позже осталось по 7—10 энтузиастов из каждого класса (двух первых и двух вторых).

Из-за большого количества детей заниматься с каждой группой удавалось только раз в неделю. Поскольку занятия проходили в дневное время, родители на них не присутствовали. Поэтому вначале работа была чрезвычайно непродуктивной: после того как мы освоили основные команды, каждый взялся за свой проект. И пока я обсуждала, что делать дальше с одним-двумя учениками, остальные сидели и ждали своей очереди, игнорируя мои призывы попробовать хоть что-то сделать самим. Фактически каждый работал только вместе со мной, а остальное время проходило в ожидании. Если взрослому можно объяснить на словах, что нужно делать, и перейти к следующему компьютеру, то для второклассника фраза: «Чтобы закрасить это окошко, нужно поднять перо, повернуть черепашку носом внутрь, зайти туда, опустить перо, взять новый цвет и покрасить» очень длинна и непонятна, даже если он знает все команды и неделю назад проделывал то же самое с крышей своего домика. Поэтому нужно было либо оставаться с ним вместо родителя и помогать проделывать все это шаг за шагом, либо заниматься с другими ребятами и возвращаться через некоторое время к ожидающим, обнаруживая, что написаны лишь 1—2 команды (в лучшем случае). С трудом давалось многим детям и понимание, что исправления в программе не приводят к исправлению рисунка, что программа — это только «рецепт» и ее надо запустить, чтобы она работала. Но были и «прирожденные программисты» — 5—6 человек (в том числе и первоклассники), которые буквально с первого дня, не зная еще команд, то и дело поглядывая на их список,

висящий на стене, сами писали программы по собственному плану. Постепенно в этом режиме стали работать все больше ребят, а к концу учебного года мне уже не приходилось вникать в каждый проект. Все работали самостоятельно.

И все же я долго еще жалела о тех временах, когда рядом с моими пяти-шестилетками сидели родители. Опыт показал, что ребята, которые занимаются вместе с ними, гораздо быстрее и легче преодолевают начальный этап, больше успевают сделать за занятие, часто приходят на занятие с уже написанной дома программой.

Наши два кружка провели олимпиаду по Лого среди детей 6—9 лет. Меня восхитила работа ребят перед олимпиадой: такого участия, такой взаимовыручки я у них еще не видела. Большинство родителей впервые узнали, как много умеют их дети. Теперь нет-нет да и зайдет на занятие чья-нибудь мама, а сын с гордостью объясняет ей, чем он сейчас занят.

Принимали мы участие и в российско-корейском проекте по Лого обмену. Но тут ребята проявили неожиданный консерватизм. Многие хотели только продолжать свои прежние проекты, украшая их крупными разноцветными надписями и музыкой, а те, кто решил начать новые, весьма своеобразно интерпретировали тему строительства общего дома на необитаемой планете, изображая либо сцену крушения космических кораблей, либо строительство дома из блоков с помощью трактора и подъемного крана, а то и роботов. Их вообще меньше волновал сюжет, нежели его техническая реализация. Как сконструировать вращающиеся колеса, составить кран из двух черепашек или сделать, чтобы трос разматывался, а потом сматывался обратно — вот это было важно! И мне не хотелось останавливать моих юных программистов, ведь они только начали открывать мир Лого. А в нем — огромные возможности, которые детям предстоит освоить.

А. Г. Юдина,

учитель информатики средней школы № 112, Москва

## ИСТОРИЯ С ЗАЙЦЕМ

Я хочу рассказать о работе над небольшим проектом в ЛогоМирах. Эта компьютерная среда позволяет, в частности, моделировать движение, реализовать несложную анимацию, программируя на языке Лого.

В ходе этой работы мы столкнулись с некоторыми затруднениями математического характера и преодолевали их не путем поиска и применения подходящей математической формулы, а экспериментальным образом. Именно возможность осуществления экспериментальной, исследовательской деятельности в компьютерной среде мне и хочется продемонстрировать.

В кружок, где предполагалась работа в среде ЛогоМиры, записались девятиклассники близнецы — Паша и Петя. Занятия начались с черепашью геометрии — одновременного освоения процедурного программирования и движения исполнителя в относительных полярных координатах. Успехи Пети и Паши были невелики, как, по-видимому, и в школе по математике. Я, можно сказать, махнула на них рукой (как на юных программистов). Братья молчаливо просиживали все занятия, потихоньку самостоятельно осваивая графические возможности среды — встроенный графический редактор и работу с формами (фигурками). Через несколько занятий я просто поразилась, какие у них стали получаться живые движущиеся фигуры, какие интересные многоцветные «декорации».

*Несколько общих соображений. Как известно, у одних людей преобладает образное мышление, у других — рационально-логическое. Первым ближе мир искусства, вторым — мир науки и техники. Наши школьные программы, начиная с младших классов, нацелены в основном на развитие рационально-логического мышления. В результате дети, которые более склонны мыслить образно, оказываются в неравном положении. Они с трудом вписываются в школьные схемы, у них мало шансов стать успевающими учениками. У таких детей часто вырабатывается стойкое отвращение к математике, физике, химии, тем более что эти науки предстают перед ними в основном как аналитические дисциплины, в которых царят формулы и строгие выводы. Задайте-ка вопрос старшекласснику, скажем, по физике. Ручаюсь, он (она) наморщит лоб и начнет подыскивать в памяти подходящую формулу. Если формула не вспомнится, никаких попыток порассуждать, найти аналогию, опереться на какие-то глобальные понятия или (хотя бы мысленно) поставить эксперимент. А ведь научное знание не только по физике, но и математическое выросло из эксперимента. Как повернуть школьное преподавание точных наук от формульной схоластики к живой экспериментальной и исследовательской учебной деятельности подростков? Тогда, может быть, трудности в освоении точных наук не были бы столь непреодолимы для тех, кого сегодня считают неспособными.*

По-видимому, Паша и Петя имеют проблемы в школе как раз потому, что у них образное мышление преобладает над рационально-логическим (по крайней мере усидчивости, аккуратности, работоспособности им не занимать). Почти на каждое занятие братья стали приходить с новыми идеями и персонажами.

Особенно мне понравился заяц, для которого Паша нарисовал полтора десятка форм. Вот некоторые из них:



Чтобы не путаться в формах, Паша придумал следующий прием: прямо в поле форм в уголке рисовал ее номер (в конце работы эти номера легко стереть).

Заяц мог двигаться по экрану в четырех направлениях: слева направо, справа налево, к зрителю и от зрителя. Набор процедур выглядел так:

это прямо

```
seth 0
setsh 36 wait 1 fd 5 setsh 46 wait 1 fd 5
setsh 51 wait 1 fd 5 setsh 47 wait 1 fd 5
end
```

это обратно

```
seth 180
setsh 49 wait 1 fd 5 setsh 50 wait 1 fd 5
setsh 53 wait 1 fd 5 setsh 52 wait 1 fd 5
end
```

это туда

```
seth -90 setsh 34 wait 1 fd 6
setsh 31 wait 1 fd 6
setsh 33 wait 1 fd 6
setsh 32 wait 1 fd 6
setsh 33 wait 1 fd 6
setsh 48 wait 1 fd 6
setsh 35 wait 1 fd 6
setsh 34 wait 1 fd 6
end
```

это сюда

```
seth 90
setsh 19 wait 1 fd 6 setsh 16 wait 1 fd 6
setsh 18 wait 1 fd 6 setsh 17 wait 1 fd 6
setsh 18 wait 1 fd 6 setsh 21 wait 1 fd 6
setsh 20 wait 1 fd 6 setsh 19 wait 1 fd 6
end
```

\* Система ЛогоМиры двуязычна, Вы можете использовать как русские, так и английские термины и даже смешивать их в одной программе. Команда **seth** (русское название — **нов\_курс**) устанавливает черепашку заданным курсом. **Setsh** является английским именем команды **нов\_форма**, **wait** — имя команды **жди**, **fd** — имя команды **вперед**.

Команда **нов\_форма** изменяет форму черепашки («одевает на нее новую форму»). В качестве параметра команде сообщается номер или имя нужной формы. В системе ЛогоМиры имеется большое количество уже готовых форм, хотя, конечно, пользователь может создавать новые и менять уже имеющиеся.

Команда **жди** вызывает паузу, время паузы определяется значением параметра.

Команда **вперед** перемещает черепашку вперед на указанное расстояние.

Слово **end** используется в тексте программы вместо русского слова **конец**.

Каждая процедура реализует небольшую пробежку зайца в заданном направлении и представляет собой последовательность фаз движения. Каждая фаза — смена формы, пауза и продвижение.

В рабочем поле Паша нарисовал «декорации» и поместил четыре кнопки — «туда», «сюда», «прямо» и «обратно». Выбирая мышкой кнопку, можно было «гонять» зайца по экрану.

Я предложила Паше попробовать реализовать игру, в которой необходимо было бы провести этого зайца по клетчатому разноцветному полю, где клетки некоторого цвета (или нескольких цветов) являлись бы ловушками. Попав на них, заяц исчезнет, провалится или еще что-нибудь в этом роде.

Сначала управление зайцем было перенесено на клавиатуру (в поле стрелок):

```
это заяц
make "q ascii readchar
if :q = 30 [прямо]
if :q = 31 [обратно]
if :q = 28 [туда]
if :q = 29 [сюда] заяц
end
```

Переменные в Лого создаются с помощью команды **пусть (make)**. Этой команде сообщается имя переменной и значение, которое следует присвоить переменной. Так, например, команда **make «скорость60»** создаст переменную с именем «скорость» и присвоит ей значение 60. Если переменная «скорость» существовала до выполнения данной команды, то новая переменная создаваться не будет, изменится значение существующей переменной. Двоеточие перед именем переменной указывает на то, что мы хотим использовать хранящееся в ней значение. Например, команда **вперед :скорость** переместит в нашем случае черепашку на 60 шагов.

\* Автор комментариев к данной статье — С. Ф. Сопрунов. — Прим. ред.

Функция жди\_символ (`readchar`) приостанавливает работу программы до тех пор, пока пользователь не нажмет какую-либо клавишу на клавиатуре. Значением функции является символ, соответствующий нажатой клавише. Функция код (`ascii`) вычисляет числовой (так называемый ASCII) код указанного символа. Например, значением выражения `ascii 'F'` будет число 70 (код буквы F).

Работа программы зайц может быть описана следующим образом: программа ждет, пока не будет нажата некоторая клавиша. После того как клавиша нажата, код, соответствующий этой клавише, записывается в переменную «q». Далее с помощью четырех команд `if` (если) этот код сравнивается с числами 28, 29, 30 и 31. Если код совпадает с одним из этих чисел (это означает, что была нажата клавиша со стрелочкой) то выполняется соответствующая программа — туда, сюда, прямо или обратно. После этого программа зайц запускает себя вновь, чтобы проанализировать следующее нажатие клавиши. Программа, в принципе, работает бесконечно.

Вместо четырех кнопок на экране оказалась одна — «заяц».

Затем мне удалось наконец донести до Пашиного сознания идею переменного параметра процедуры. При перемещении зайца повторяется все время один и тот же набор действий, меняется только номер формы. Если описать процедуру смены формы (сф), то весь набор процедур упрощается:

```
это сф :n
setsh :n wait 1 fd 5
end
```

```
это прямо
seth 0 сф 36 сф 46 сф 51 сф 47
end
```

```
это обратно
seth 180 сф 49 сф 50 сф 53 сф 52
end
```

```
это туда
seth -90 сф 34 сф 31 сф 33 сф 32 сф 33 сф 48 сф 35 сф 34
end
```

```
это сюда
seth 90 сф 19 сф 16 сф 18 сф 17 сф 18 сф 21 сф 20 сф 19
end
```

Следующая задача была такая: движение будет происходить в пространстве. Удаляясь от зрителя, заяц должен с каждым шагом уменьшаться в размерах, должен также уменьшаться и его шаг. При движении к зрителю размер и шаг будут соответственно увеличиваться, а при перемещении вправо-влево — оставаться неизменными. Значит, размер и шаг должны стать переменными величинами, пусть ими управляет одна и та же переменная `s`. Тогда возникает вопрос: а как будет изменяться переменная `s`? Первоначальное предположение было такое: с каждой фазой она будет уменьшаться (увеличиваться) на одно и то же число, например на единицу.

После десятка проб были подобраны коэффициенты, связывающие размер и шаг зайца с переменной `s`. Процедура смены формы «расщепилась» на три — для удаления и приближения зайца (при смене формы переменная `s` должна соответственно уменьшаться или увеличиваться) и для перемещений вправо-влево. Набор процедур принял следующий вид:

```
это сф :n
setsh :n wait 1 fd :s / 3
end
```

```
это сф1 :n
setsh :n wait 1 fd :s / 3 setsize 2 * :s make "s :s - 1
end
```

```
это сф2 :n
make "s :s + 1 setsh :n wait 1 fd :s / 3 setsize 2 * :s
end
```

```
это прямо
seth 0 сф1 36 сф1 46 сф1 51 сф1 47
end
```

```
это обратно
seth 180 сф2 49 сф2 50 сф2 53 сф2 52
end
```



Выражение `n` в заголовке программы `сф` означает, что в программе используется параметр, значение которого должно быть указано при вызове программы, например `сф 18`.

Использование знаков арифметических действий в Лого обычно не вызывает затруднений, разве что знак умножения выглядит слегка непривычно — «\*».

Команда `нов_размер (setsize)` устанавливает размер черепашки.

Функция `форма (shape)` сообщает номер формы, одетой на черепашку.

Команда `по (pd)` опускает перо черепашки, команда `пп (pu)` — поднимает перо. Команда `штамп (stamp)` «штампует» форму черепашки на листе.

Остальные процедуры не изменились.

Перед каждой очередной пробой зайца «перетаскивали» мышкой на старт и выполняли начальные установки:

```
make "s 40 setsize 80
```

После того как заяц уже начал вполне правдоподобно перемещаться в перспективе, возникла необходимость как-то отслеживать его положения на экране после каждого нажатия на клавиши со стрелками. Для этого была написана процедура, которая заставляла зайца оставлять метку (точку) после каждого перемещения:

```
это точ
make "z shape setsh 60 pd stamp pu setsh :z
end
```

Вызовы этой процедуры были добавлены в соответствующие места главной процедуры:

```
это заяц
make "q ascii readchar
if :q = 30 [прямо точ]
if :q = 31 [обратно точ]
if :q = 28 [туда точ]
if :q = 29 [сюда точ]
заяц
end
```

Теперь на экране можно было увидеть всю «историю» перемещений зайца:



В результате возникли новые проблемы.

Во-первых, очевидно, что число шагов по самой дальней горизонтали должно быть равно числу шагов на переднем плане. Другими словами, курс зайца при перемещениях «прямо» и «обратно» должен меняться, линии движения должны сходиться где-то на горизонте. Исходя из размеров поля, определили, что этих линий будет шесть. Чтобы заяц мог «знать», на какой из этих линий он в данный момент находится, ввели переменную `i` — номер линии (вначале она устанавливается равной единице, а затем при каждом нажатии стрелок вправо или влево соответственно уменьшается или увеличивается). Чтобы заяц «не выскакивал» за эти шесть линий, в главную процедуру были внесены ограничители:

```
это заяц
make "q ascii readchar
if :q = 30 [прямо точ]
if :q = 31 [обратно точ]
if :q = 28 [if :i < 6 [туда точ make "i :i + 1 ]]
if :q = 29 [if :i > 1 [сюда точ make "i :i - 1 ]]
заяц
end
```

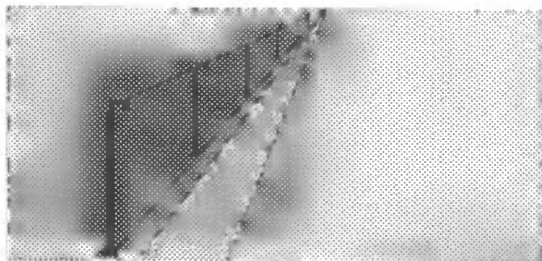
На этом этапе более четко определилась основная задача — добиться того, чтобы заяц при перемещении по полю выбранного размера 6×6 все время попадал на одни и те же 36 точек. (А на деле он все время смещался и точки-метки «расползались» и постепенно заполняли экран.) Чтобы заяц был ограничен и шестью горизонталями, пришлось ввести также и номер горизонтали  $j$ , и соответствующие ограничители:

```

это заяц
make "q ascii readchar
if :q - 30 [if :j < 6 [прямо точ make "j :j + 1]]
if :q - 31 [if :j > 1 [обратно точ make "j :j - 1]]
if :q - 28 [if :i < 6 [туда точ make "i :i + 1]]
if :q - 29 [if :i > 1 [сюда точ make "i :i - 1]]
заяц
end

```

Второй проблемой оказалось уточнение закона изменения переменной  $s$ . С появлением на экране «привязки» стало очевидно, что расположение точек и изменение размера зайца при удалении/приближении все-таки выглядят не вполне естественно. После рисования перспектив домов или телеграфных столбов, уходящих вдаль, был признан правильным метод умножения переменной  $s$  на некоторый коэффициент, меньший единицы, при удалении зайца и деления  $s$  на этот же коэффициент при приближении зайца.



Линейные размеры при изображении перспективы образуют геометрическую прогрессию, а не арифметическую.

Методом проб и ошибок были заново подобраны коэффициенты, таким же опытным путем были определены углы наклона линий движения:

```

это сф :n
setsh :n wait 1 fd :s / 9
end

```

```

это сф1 :n
setsh :n wait 1 fd :s / 5 setsize round :s make "s :s * :k
end

```

```

это сф2 :n
make "s :s / :k setsh :n wait 1 fd :s / 5 setsize round :s
end

```

Функция округли (**round**) округляет указанное число. Команда **нов\_поз** (**setpos**) устанавливает черепашку в заданную точку листа, точка задается с помощью координат, координаты центра листа — (0, 0).

```

это прямо
if :i - 1 [seth -26]
if :i - 2 [seth -15]
if :i - 3 [seth -5]
if :i - 4 [seth 5]
if :i - 5 [seth 15]
if :i - 6 [seth 26]
сф1 36 сф1 46 сф1 51 сф1 47
end

```

```

это обратно
if :i - 1 [seth 154]
if :i - 2 [seth 165]
if :i - 3 [seth 175]

```

```

if :i = 4 [seth -175]
if :i = 5 [seth -165]
if :i = 6 [seth -154]
сф2 49 сф2 50 сф2 53 сф2 52
end

```

Начальные установки приняли следующий вид:

```
setpos [130 -120] make "i 1 make "j 1 make "s 80 setsize :s make "k 0.96
```

После этих модернизаций на экране была получена картинка, которая хорошо иллюстрирует следующую возникшую проблему:



Горизонтальные ряды точек все больше «выгибаются», удаляясь от переднего края. Значит, каждое перемещение зайца по направлениям «прямо» — «обратно» также зависит от номера линии  $i$ . Чем ближе к центру, тем шаг должен быть меньше. Конечно, тут можно было бы спросить: «Тригонометрию проходил?» и подсказать готовую формулу с синусом. Однако полезно сначала дать ученикам «прочувствовать» практическую значимость соответствующих зависимостей, после чего и формулы будут восприниматься иначе.

Таким образом, появилась еще одна переменная  $m$ , и после нескольких экспериментов набор процедур принял окончательный вид.

```

это заяц
t1, setpos [130 -120] make "i 1 make "j 1 make
"s 80 setsize :s make "k 0.96
упр
end

```

```

это упр
make "q ascii readchar
if :q = 30 [if :j < 6 [прямо make "j :j + 1]]
if :q = 31 [if :j > 1 [обратно make "j :j - 1]]
if :q = 28 [if :i < 6 [туда make "i :i + 1]]
if :q = 29 [if :i > 1 [сюда make "i :i - 1]]
упр
end

```

```

это сф :n
setsh :n wait 1 fd :s / 9
end

```

```

это сф1 :n
setsh :n wait 1 fd :s / :m setsize round :s make "s :s * :k
end

```

```

это сф2 :n
make "s :s / :k setsh :n wait 1 fd :s / :m setsize round :s
end

```

```

это прямо
if :i - 1 [seth -25 make "m 4.75]
if :i - 2 [seth -16 make "m 5.0]
if :i - 3 [seth -5 make "m 5.2]
if :i - 4 [seth 5 make "m 5.2]
if :i - 5 [seth 16 make "m 5.0]
if :i - 6 [seth 25 make "m 4.75]
сф1 36 сф1 46 сф1 51 сф1 47
end

```

```

это обратно
if :i - 1 [seth 155 make "m 4.75]
if :i - 2 [seth 164 make "m 5.0]
if :i - 3 [seth 175 make "m 5.2]
if :i - 4 [seth -175 make "m 5.2]
if :i - 5 [seth -164 make "m 5.0]
if :i - 6 [seth -155 make "m 4.75]
сф2 49 сф2 50 сф2 53 сф2 52
end

```

```

это туда
seth -90 сф 34 сф 31 сф 33 сф 32 сф 33 сф 48 сф 35 сф 34
end

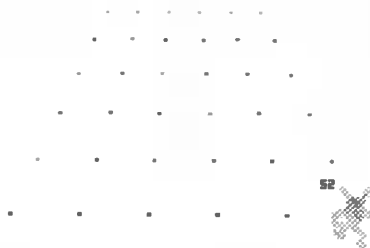
```

```

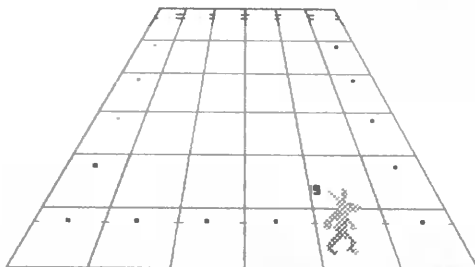
это сюда
seth 90 сф 19 сф 16 сф 18 сф 17 сф 18 сф 21 сф 20 сф 19
end

```

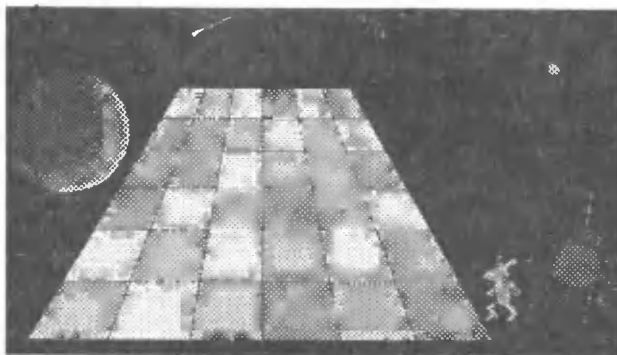
После того как заяц, наконец, стал «приземляться» в одних и тех же точках и «точечное» поле приняло показанный на рисунке вид, вспомогательная процедура «точ» была выброшена.



С помощью встроенного графредактора по основе точечного поля было начерчено настоящее клетчатое, а точки стерли.



После этого Паша нарисовал вокруг поля красивые «космические» декорации.



Таким образом, получилась основа игры. Затем уже дело фантазии — запрограммировать различные приключения зайца при попадании на клетки определенного цвета. Для этого в процедуру «упр» можно вставить строчки типа:

```
if colorunder = 65 [упал stop]
if colorunder = 75 [пропал stop]
if colorunder = 85 [взлетел stop]
```

.....

Функция **цвет\_поля (colorunder)** сообщает номер цвета экрана в том месте, где находится черепашка.

А теперь давайте подсчитаем, со сколькими математическими понятиями вы можете познакомить своих учеников в процессе работы над подобным проектом: переменная величина, параметр, массив, индекс элемента, арифметическая и геометрическая прогрессии, прямая пропорциональность, подобие геометрических фигур (при рассмотрении закономерностей изменения линейных размеров в перспективе), симметрия, соотношения углов, тригонометрические функции и зависимости.

Наибольшую пользу, как мне кажется, извлекут именно слабые в учебе подростки. Конечно, большинство юных хакеров (поголовно имеющих ярко выраженный «аналитический ум») достаточно легко перешагнут через все эти синусы-арксинусы. Да только художественные наклонности обнаруживаются у них редко. Как в сказке — либо дудочка, либо кувшинчик. (Вот он — глубокий непроходимый овраг между «гуманитариями» и «математиками», который старательно расширяют школьные программы.)

Таким образом, все члены семейства компьютерных Лого-сред — LogoWriter, MicroWorlds (ЛогоМиры), Lego-Logo — прекрасные инструменты для учителя. В первую очередь они позволяют воплотить педагогический принцип: учиться, делая что-то, создавая что-либо нужное или интересное. Может быть, наиболее привлекательная возможность в такой работе — не разделять искусственными границами различные области деятельности (например, рисование, мультипликацию и математику, как это было в описанном проекте).

**А. Л. Горбунов,**

*преподаватель УВК № 1801, Москва*

## **КОМПЬЮТЕРНЫЕ СЕТИ ДЛЯ ДЕТЕЙ ЧЕРЕЗ LEGO TC LOGO**

### **1. Цели проекта**

Во все времена сети весьма широко использовались людьми. Перечень сетей для ловли рыбы, зверей, гамаки в XIX — начале XX в. дополнился телефонными сетями. Наше время с полным правом можно назвать эпохой компьютерных сетей или, заглядывая в будущее, глобальных цифровых сетей передачи информации (информационные супермагистраль). С помощью компьютерных сетей можно серьезно повысить «уловы» требуемых данных в море информации, окружающем современного человека, и увеличить эффективность решения многих проблем. Однако практика показывает, что непонимание принципов организации и специфики работы компьютерной телекоммуникации зачастую бывает серьезным препятствием для полноценного использования сетевых возможностей. Это обстоятельство диктует целесообразность осуществления проекта создания небольшой компьютерной сети с помощью комплекса типа Lego-Logo детьми младшего школьного возраста, который позволяет довести до них содержание ряда первичных понятий, касающихся некоторых аспектов создания и эксплуатации сети обмена цифровыми данными. Проект приемлем для реализации учащимися 9—10 лет, имеющими опыт работы с Lego TC Logo или Control Lab.

Основные темы проекта:

- сеть;
- кодирование;
- протоколы;
- сетевые приоритеты.

### **2. Минимальный комплект необходимых технических средств:**

- персональный компьютер (IBM PC или Apple с соответствующей операционной системой);
- комплект Lego TC Logo.

### **3. Эскизный проект Lego-сети**

Lego TC Logo содержит все необходимое для создания небольшой и несложной, но совершенно настоящей компьютерной сети: язык программирования для разработки программного обеспечения управления сетью; датчики, моторы и лампы для построения удаленных терминалов (устройства, через которые можно воспользоваться услугами сети); компьютер, который можно использовать как сервер (центральный компьютер, управляющий обменом в сети и хранящий большие объемы информации, доступные для совместного использования абонентами), соединительные провода для физической связи между абонентами сети.

Начните предварительную проработку проекта сети с объяснения значения незнакомых слов (терминал, сервер). Обсудите с детьми схему, приведенную на рисунке: от кого и кому можно передавать сообщения в такой сети? Важно, чтобы юные разработчики поняли, что диспетчером для всех передаваемых сообщений является сервер. Следующая тема для разговора — каким образом с помощью электрических Lego-элементов можно создать терминал. Датчики касания — кнопки применимы в качестве устройств для отправки сообщений, лампы и моторы — для приема. Затем, естественно, встает вопрос о кодировании передаваемых сообщений. Предложите детям обдумать доступные способы кодирования (количество импульсов-нажатий на кнопку, длительность импульсов-нажатий, комбинации первого и второ-

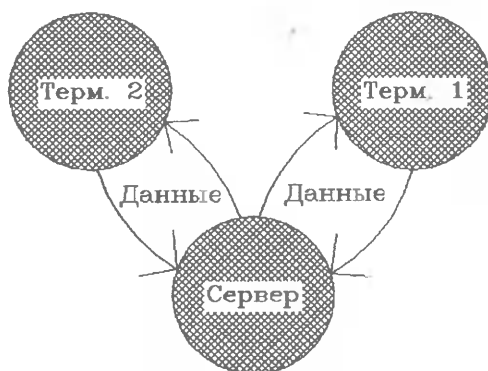


Схема Lego-сети

го), их достоинства и недостатки с точки зрения возможного количества передаваемых сообщений, простоты передачи и расшифровки. Если применяется (как предлагается далее) наиболее простой число-импульсный метод, то каждое сообщение связывается с определенным числом нажатий кнопки-датчика при передаче и всплешек лампы (или числом оборотов мотора) при приеме.

Представляется очень интересным исследовать вместе с учащимися вопрос о том, как одно сообщение будет отделяться от другого. Это хорошая возможность разобраться с емким понятием протокола, т. е. набора правил, которые позволяют организовать корректный обмен данными между абонентами сети (серверами и терминалами). Другими словами, протоколы позволяют абонентам правильно «понимать» друг друга. Для создания сети детям понадобится разработать что-то вроде DATA LINK PROTOCOL из телекоммуникационной модели, предусматриваемой спецификацией OSI Международной организации стандартов (ISO). Необходимо определить:

- а) как терминал информирует сервер о передаче сообщения;
- б) как сервер подтверждает нормальный прием сообщений;
- в) как сервер извещает терминал о посылке сообщения;
- г) что означает каждое сообщение.

Очень интересно отдельно рассмотреть вопрос о фиксации и расшифровке принимаемых сообщений (приводимое ниже программное обеспечение не решает эти задачи). Вся активность учащихся во время предварительного обсуждения проекта (варианты решения формулируемых проблем, принимаемые к исполнению способы) должна фиксироваться в тетрадях (бумажных или электронных) — эти записи могут понадобиться позднее при модификации и совершенствовании создаваемого продукта.

#### Вопросы для дискуссии.

1. В чем заключается сходство между рыбацкой и компьютерной сетями?
2. Почему компьютерные сети так важны сегодня?

### 4. Реализация проекта

Разработка программного обеспечения управления Lego-сетью может оказаться достаточно трудной задачей не только для детей (конечно все зависит от уровня «программистской квалификации» учащихся, которая может сильно различаться). Поэтому ниже приводится листинг готовой «сетевой операционной системы» LegoNet, содержащей 12 Logo программ (версия для MS DOS, LEGONET является главной программой). Возможно, более реальной и интересной образовательной задачей будет не разработка, а исследование и совершенствование LEGONET.

В нашей реализации терминал состоит из Lego-датчика касания (кнопки) и

Lego-лампочки. Электрическое оборудование терминала 1 подключено к портам 1 и 6 интерфейсного блока (Lego-интерфейс А), а терминала 2 — к портам 2 и 7. После запуска программа LegoNet выводит на экран мигающую строку «Lego TC Logo Network!». Мигание сопровождается короткими звуковыми сигналами — сервер находится в режиме «Готов к обмену». Рассмотрим процедуры, которые позволяют осуществлять LegoNet.

#### 4.1. Посылка сообщений от одного терминала другому.

Нажмите кнопку на терминале — отправителе сообщений. Если сервер в данный момент не занят, он ответит 5 короткими вспышками лампочки терминала-отправителя, сопровождаемыми звуковыми сигналами. Это означает «Готов принять сообщение». Затем в течение 3 секунд можно послать сообщения — от 1 до 5 нажатий кнопки. Что означает каждое сообщение, решают дети. Если вы не нажмете кнопку ни разу, сервер восстановит режим готовности.

В случае посылки вами какого-либо сообщения информация о нем появляется на экране и лампочка на терминале-приемнике дважды зажигается двумя короткими вспышками, сопровождаемыми звуковыми сигналами и разделенными паузой — сигнал «Примите сообщение от терминала». Затем после небольшой паузы лампочка на терминале-приемнике зажигается длинными интервалами (0.5 секунды) в количестве, соответствующем посланному сообщению.

Далее процедура повторяется до тех пор, пока вы не перестанете нажимать кнопку на терминале-отправителе.

#### 4.2. Посылка сообщений от терминала серверу.

Данная процедура работает так же, как описанная в п. 4.1, однако количество нажатий кнопки на терминале-отправителе должно быть больше 5 (это признак обращения к серверу). В данном случае на экране появляется строка «Message from TERM-1 to SERVER :» (или «from TERM-2») и число, соответствующее количеству нажатий кнопки на терминале-отправителе.

#### 4.3. Посылка сообщений от сервера к терминалу.

Нажмите любую клавишу на клавиатуре сервера, когда он находится в режиме «Готов к обмену». Компьютер ответит звуковым сигналом низкой тональности, и на экране появится приглашение ввести номер терминала — приемника сообщений. Введите нужную цифру (1 или 2).

Компьютер выдаст серию из пяти коротких звуковых сигналов и приглашение ввести код сообщения (цифра от 1 до 5). Чтобы прервать сеанс обмена с терминалом, нажмите клавишу «s».

В случае посылки вами какого-либо сообщения информация о нем появляется на экране и лампочка на терминале-приемнике трижды зажигается двумя короткими вспышками, сопровождаемыми звуковыми сигналами и разделенными паузой — сигнал «Примите сообщение от сервера». Затем после небольшой паузы лампочка на терминале-приемнике зажигается длинными интервалами (0.5 секунды) в количестве, соответствующем посланному сообщению.

Далее процедура повторяется до тех пор, пока вы не нажмете клавишу «s».

## 5. Программное обеспечение LEGONET

5.1. LEGONET — главная программа в пакете программной поддержки Lego-сети. LEGONET выводит на экран заставку «Lego TC Logo NetWork !», предварительно очистив экран и проверяет наличие запросов на обмен от терминалов и с клавиатуры сервера (операторы IF). В случае обнаружения таковых управление передается соответственно подпрограммам TERM-1, TERM-2 и SERVER. При отсутствии запросов программа перезапускается.

```
TO LEGONET
CG HT PU SETPOS [-90 0]
LABEL [LEGO TC logo NETWORK !]
IF KEY? = "TRUE [SERVER]
LTO [6 7]
MAKE "T1 FIRST COUNTER
MAKE "T2 LAST COUNTER
```



```

IF :T1 > 0 [TERM-1]
IF :T2 > 0 [TERM-2]
RESETC
TONE 300 2 WAIT 2
LEGONET
END

```

5.2. SERVER — программа определения направления передачи сообщения от сервера. Выводит на экран приглашение ввести номер терминала-приемника «To TERMINAL # (1 or 2):»; в зависимости от ответа делает активным порт 1 или 2 и передает управление программе SEND-S.

```

TO SERVER
TONE 150 10
TYPE [To TERMINAL # (1 or 2):]
MAKE "W READLISTCC
CC
IF :W - [1] [TTO 1 SEND-S]
IF :W - [2] [TTO 2 SEND-S]
END

```

5.3. SEND-S — программа, обеспечивающая посылку сообщений от сервера терминалам. Генерирует серию звуковых сигналов, выводит на экран приглашение ввести код сообщения «What to send? (1..5, Stop — s):». После ввода пользователем кода сообщения передает его терминалу, сопровождая передачу сигналом «Примите сообщение от сервера» (подпрограмма ACK-SS), и вновь выводит приглашение. При прерывании сеанса обмена (ввод символа «s») возвращает управление программе LEGONET.

```

TO SEND-S
REPEAT 5 [TONE 500 1 WAIT 1]
TYPE [What to send? (1..5, Stop — s):]
MAKE "S READLISTCC
CC
ACK-SS
IF :S - [s] [LEGONET]
IF :S - [1] [WAIT 3 ONFOR 5]
IF :S - [2] [REPEAT 2 [WAIT 3 ONFOR 5]]
IF :S - [3] [REPEAT 3 [WAIT 3 ONFOR 5]]
IF :S - [4] [REPEAT 4 [WAIT 3 ONFOR 5]]
IF :S - [5] [REPEAT 5 [WAIT 3 ONFOR 5]]
WAIT 3
SEND-S
END

```

5.4. TERM-1 — программа реакции на запрос обмена от терминала 1. Очищает экран, посылает терминалу сигнал «Готов принять сообщение» (подпрограмма ACK-R), в течение 3 секунд ожидает сообщений от терминала 1. При наличии сообщений управление передается программе SEND1, при отсутствии — управление возвращается программе LEGONET.

```

TO TERM-1
CG PU HT HOME
TTO 1
ACK-R
RESETC
LTO 6
WAIT 30
MAKE "C COUNTER
IFELSE :C = 0 [LEGONET] [SEND1]
END

```

5.5. SEND1 — программа, обеспечивающая посылку сообщений от терминала 1. Определяет направление передачи (первый оператор IF) по количеству импульсов в сообщении. При числе импульсов больше 6 передает управление программе SEND1-S. В противном случае выводит на экран код сообщения от терминала 1 терминалу 2, посылает терминалу 2 сигнал «Примите сообщение от терминала» (подпрограмма ACK-S) с последующим сообщением и передает управление программе TERM-1.

```

TO SEND1
SETPOS [-150 80]
IF :C > 5 [SEND1-S]
LABEL [Message from TERM-1 to TERM-2 :]
BK 20
LABEL :C
TTO 2
ACK-S
IF :C - 1 [WAIT 3 ONFOR 5]
IF :C - 2 [REPEAT 2 [WAIT 3 ONFOR 5]]
IF :C - 3 [REPEAT 3 [WAIT 3 ONFOR 5]]
IF :C - 4 [REPEAT 4 [WAIT 3 ONFOR 5]]
IF :C - 5 [REPEAT 5 [WAIT 3 ONFOR 5]]
WAIT 3
TERM-1
END

```

5.6. SEND1-S — программа, обеспечивающая индикацию сообщения от терминала 1 серверу. Выводит на экран код сообщения и передает управление программе TERM-1.

```

TO SEND1-S
LABEL [Message from TERM-1 to SERVER :]
BK 20
LABEL :C
WAIT 15
TERM-1
END

```

5.7. TERM-2 — программа реакции на запрос обмена от терминала 2. Очищает экран, посылает терминалу сигнал «Готов принять сообщение» (подпрограмма ACK-R), в течение 3 секунд ожидает сообщений от терминала 2. При наличии сообщений управление передается программе SEND2, при отсутствии — управление возвращается программе LEGONET.

```

TO TERM-2
CG PU HT HOME
TTO 2
ACK-R
RESETC
LTO 7
WAIT 30
MAKE "C COUNTER
IFELSE :C = 0 [LEGONET] [SEND2]
END

```

5.8. SEND2 — программа, обеспечивающая посылку сообщений от терминала 2. Определяет направление передачи (первый оператор IF) по количеству импульсов в сообщении. При числе импульсов больше 6 передает управление программе SEND2-S. В противном случае выводит на экран код сообщения от терминала 2 терминалу 1, посылает терминалу 1 сигнал «Примите сообщение от терминала» (подпрограмма ACK-S) с последующим сообщением и передает управление программе TERM-2.

```

TO SEND2
SETPOS [-150 80]
IF :C > 5 [SEND2-S]
LABEL [Message from TERM-2 to TERM-1 :]
BK 20
LABEL :C
TTO 1
ACK-S
IF :C - 1 [WAIT 3 ONFOR 5]
IF :C - 2 [REPEAT 2 [WAIT 3 ONFOR 5]]
IF :C - 3 [REPEAT 3 [WAIT 3 ONFOR 5]]
IF :C - 4 [REPEAT 4 [WAIT 3 ONFOR 5]]
IF :C - 5 [REPEAT 5 [WAIT 3 ONFOR 5]]
WAIT 3
TERM-2
END

```

5.9. SEND2-S — программа, обеспечивающая индикацию сообщения от терминала 2 серверу. Выводит на экран код сообщения и передает управление программе TERM-2.

```
TO SEND2-S
LABEL [Message from TERM-2 to SERVER :]
BK 20
LABEL :C
WAIT 15
TERM-2
END
```

5.10. ACK-S — программа выдачи сигнала «Примите сообщение от терминала».

```
TO ACK-S
REPEAT 2 [ONFOR 1 TONE 600 1 ONFOR 1
TONE 600 1 WAIT 3]
END
```

5.11. ACK-R — программа выдачи сигнала «Готов принять сообщение».

```
TO ACK-R
REPEAT 5 [TONE 500 1 ONFOR 1]
END
```

5.12. ACK-SS — программа выдачи сигнала «Примите сообщение от сервера».

```
TO ACK-SS
REPEAT 3 [ONFOR 1 TONE 600 1 ONFOR 1
TONE 600 1 WAIT 3]
END
```

## 6. Что еще?

Решения проблемы фиксации сообщения на терминале — обширное поле деятельности конструкторского и программного характера. Вариантом решения может быть запись на бумажной ленте сигналов, закодированных азбукой Морзе. В этом случае необходимо ее изучить, построить лентопротяжный аппарат при помощи Lego-моторов и программно синхронизировать его работу с передачей сигналов. От азбуки Морзе совсем недалеко до современных цифровых методов передачи данных, которые начинают вытеснять аналоговые системы, доминировавшие в аппаратуре связи с начала XX в., — техника совершила диалектический виток. Передачу информации в LegoNet можно организовать по примеру известного интерфейса RS-232, попутно изучив двоичное кодирование символов, пакетную передачу, обрамление пакетов служебными сигналами.

Другой интересный вопрос — расшифровка сообщений. Сообщения, принимаемые сервером, расшифровать и выводить на экран в текстовом виде несложно.

Работа с LegoNet естественным образом подведет учащихся к проблеме конфликтов между абонентами сети за обладание ее ресурсами. Это хорошая возможность поговорить о понятии «система сетевых приоритетов». Что может служить основанием для назначения того или иного приоритета определенному абоненту? Каким образом дополнить программы LegoNet, чтобы возможно было назначать и изменять приоритеты?

## Литература

1. Lego TC Logo. Reference guide. (1989) Logo Computer Systems Inc. Pointe-Claire, Quebec, Canada.
2. Krol E. (1992) The Whole Internet. ISBN 1-56592-025-2, O'Reilly & Associates, USA.
3. McCain T. & Ekelund M. (1993) Computer Networking for Educators. ISBN 1-56484-039-5, ISTE, Eugene, OR, USA.
4. Мячев А. А. и др. Интерфейсы систем обработки данных: Справочник. М.: Радио и связь, 1989.

**О. В. Митина,**

*кандидат психологических наук, МГУ*

## ПРОГРАММНАЯ СРЕДА ЛОГО ГЛАЗАМИ ПСИХОЛОГА

— Видите ли, — смущенно развел руками профессор, — это моя оплошность. Я не предусмотрел в Электронике чувства. Я думал, что от них он может перегореть. Как видно, я ошибался... В нем должны быть смех, и улыбки, и веселье.

*Е. С. Велтистов.*

*«Электроник — мальчик из чемодана»*

Программная среда Лого (ЛогоМиры) была разработана и реализована под руководством американского психолога С. Пейпета в 1989 г. в Массачусетском технологическом институте. Она была создана не просто как формализованный язык программирования, а как среда, в которой дети могли бы научиться естественному общению с компьютером. Главным персонажем и партнером ребенка по обучению и общению здесь является компьютерная черепашка, исполняющая его команды. Ребенок имеет возможность не только следить за движением абстрактного рисунка на экране компьютера, но и собрать черепашку и другие самые разнообразные объекты из детского конструктора Лего и управлять ими с помощью компьютера [4]. В настоящее время русифицированная в Институте новых технологий образования среда Лого входит в число обязательных программных средств, которыми московские школы-новостройки обеспечивают по решению Московского департамента образования.

\*\*\*

Стремление превратить компьютер в привычного партнера по общению тесно связано с поисками исторически новых форм передачи и обработки информации. Так, например, компьютерные игры позволяют моделировать управляемую реальность. Однако здесь возникают препятствующие обстоятельства.

Во-первых, устройство должно быть хорошо знакомым и простым в управлении, чтобы каждый человек мог им воспользоваться. На заре развития автомобилеводения (в 20—30-е гг.) практически каждый владелец автомобиля мог его собрать и разобрать, старые книги по-

льны описаний деталей машин и подробного их устройства. О современной технике неискушенным пользователям зачастую известно лишь, где она включается. Такая отчужденность компьютера неизбежно должна влиять на наше сознание, если мы хотим превратить его в партнера по общению («соОБЩника»). Машина, когда ее устройство оказывается непонятным, вызывает мистический страх.

Во-вторых, язык, понятный компьютеру, должен быть прост и доступен ребенку.

Использование конструктора Лего в каком-то смысле позволяет ребенку «войти в виртуальный мир компьютера» [7] (или принять компьютер в свой мир). Объект, собранный самим ребенком, — это привычный атрибут его мира. Процесс управления игрушечной машинкой или паровозиком гораздо более «материально осязаем». С другой стороны язык, используемый в Лого, максимально приближен к естественному. На первом этапе словарь необходимых для работы команд составляется взрослым, дальше ученик делает это сам.

Проблема динамики психического развития в условиях компьютеризации в настоящее время лишь начинает разрабатываться. Так, в работе Е. И. Машбиц и др. [6] указаны всевозможные аспекты для анализа различных форм и видов взаимодействия, а также разбираются те из них, которые, по мнению авторов, удастся реализовать в условиях работы с компьютером.

Основными функциями межличностного взаимодействия являются информационно-коммуникативные, регуляционно-коммуникативные, аффективно-коммуникативные. В реальной жизни

все три функции, как правило, тесно связаны. При традиционном взаимодействии человека с компьютером основной упор делается на реализацию лишь первых двух. Аффективно-коммуникативная функция, связанная с выражением и передачей эмоций и переживаний, реализуется через систему поощрений-порицаний, однако делается это обычно в нейтральной манере. Эмоциональное обращение к пользователю применяется недостаточно, причем часто с нарушением принципов общения.

Согласно точке зрения, представленной в области психологического воздействия, называемого нейролингвистическим программированием, выделяются визуальная, акустическая и кинестетическая составляющие любого информационного сообщения [2]. Поэтому важной является проблема организации визуального и звукового рядов предъявляемой информации. По мнению В. Ф. Петренко [10], эмоционально-образный код слова можно назвать генетически первичной формой обобщения и репрезентации мира субъекту. Моделирование эмоций в этой концепции, связанной с представлением словарных связей на более глубинном семантическом уровне в форме метафор, метонимии, синестезии, реализуется через редукцию более поверхностного понятийного (на лингвистическом уровне) содержания слова к этому коду.

К особенностям Лого, позволяющим решить эту задачу, относятся комплексность предъявления информации, проявляющаяся в одновременном использовании звука, цвета и движения объекта, который может быть управляем самим ребенком. Здесь актуальны вопросы, связанные с использованием мультимедиа, музыкального сопровождения и т. д. Кинестетическая составляющая включает в себя регулируемый временной темп предъявления информации, а также строится на основе использования тактильных и кинестетических ощущений. Использование компьютерной мыши и Лего-игрушек тоже актуализирует кинестетическую модальность.

Указанные мультимедиальные возможности Лого открывают принципиально новые, не имеющие аналогов, формы работы и непосредственно психолога с ребенком.

Известно, что проективные методы являются наиболее эффективными средствами в познании личности испытуемо-

го [12]. Используя при этом образ воображения в качестве опосредующего звена, они позволяют в достаточно короткий срок получить материал, являющийся базой глубокого и всестороннего познания личности, и одновременно с этим решают психотерапевтические и психокоррекционные задачи. Одним из наиболее распространенных классов проективных методов являются рисуночные методики [13].

Будучи предельно простой для освоения и ребенком, и взрослым, среда Лого позволяет расширить используемый для различных форм психотерапевтической работы инструментарий, отказаться от сложившегося у психологов стереотипа использования компьютера лишь для автоматизированной психодиагностики с помощью опросников.

Использование цвета, звука, движения объектов на экране компьютера позволяет перевести рисуночные методики на качественно новый уровень «мультипликационных» методик. Расширяются возможности интерпретации, создается более точный инструмент для диагностики.

Так, например, в случае выполнения стандартной бланковой методики «Рисунк семьи» предполагается, что ребенок должен нарисовать на листе бумаги свою семью, и этот рисунок служит материалом для работы психолога в ходе психологической консультации. Рисунок интерпретируется по стандартной заданной схеме [13] и помогает консультанту выявить наиболее проблемные в психологическом плане моменты, касающиеся восприятия ребенком своей семьи, отношения к ней, осознания себя и своих родных. Рисование семьи в среде Лого с изменением параметров: движения, цвета, звука — не только служит средством психодиагностики семейных отношений, но и дает основу для психокоррекционной деятельности психолога: моделирования и разрешения конфликтных ситуаций, внесения игрового компонента в проблемную ситуацию и т. д. Ребенок создает на экране компьютера мультфильм о своей семье. Творческая деятельность по значимой для него проблеме позволяет объективировать эту проблему, облегчить ее разрешение. Особенно эффективным является процесс озвучивания персонажей, созданных на экране. Достижимый при этом психотерапевтический эффект сопоставим с ре-

зультатами психодраматического воздействия [5].

Унификация методики и возможность использования компьютера как в ходе тестирования (ребенок создает свой проект, работая в среде Лого), так и на этапе интерпретации существенно облегчают работу психотерапевта. Особенно это важно для школьного психолога, имеющего возможность таким образом провести предварительное обследование всех детей, не акцентируя на нем их внимания с самого начала. Ведь известно, к сожалению, что дети не любят ходить к школьному психологу, считая это зазорным, наносящим ущерб их репутации среди сверстников\*.

\*\*\*

Если рассматривать межличностное взаимодействие под углом совместной деятельности, то можно выделить такие функции: организация совместной деятельности, формирование и развитие межличностных отношений, познание людьми друг друга.

Основой серьезного возражения против тотального включения компьютеров в содержание школьного обучения является то обстоятельство, что компьютеризированное обучение в значительной степени наносит ущерб человеческому общению ребенка с учителем и со сверстниками, поскольку все больше и больше времени ученики начинают проводить за компьютером, что крайне обедняет их коммуникационные связи с точки зрения традиционного, но так необходимого для формирования полноценной личности общения, играющего в жизни подростка важную роль. В этот период межличностное общение, по мнению ряда авторов [6], вообще является ведущей деятельностью, в русле которой происходит становление основных новообразований данного возраста. Однако традиционно считается, что межличностные отношения и межличностное познание, являющиеся важнейшим компонентом социальной жизни, формируются только в контактах между людьми.

В этой связи разработчики Лого предусматривают не просто его использова-

ние в учебном процессе, но и построение на этой основе «единого учащегося сообщества» [8]. Дети группами, классами, даже целыми школами разрабатывают совместные междисциплинарные проекты. Учатся ставить и решать задачи вместе, достигать понимания друг с другом, а не только с компьютером. В качестве примера здесь можно упомянуть совместный российско-корейский проект, во многом решающий проблемы общения даже в межнационального масштабе [9].

Известный психолог Э. Берн [1], разбирая всевозможные ситуации межличностного общения, выделяет три основные стереотипные ролевые позиции, которые партнеры по общению могут занимать. Это позиция «родителя» — доминантная позиция сверху вниз. Человек, занимающий эту позицию, ожидает, что партнер будет ему подчиняться и выполнять все его распоряжения (будет «послушным ребенком») или откажется вести себя так, как от него требует «родитель», при этом не отжуждяя поиском конструктивных компромиссов («непослушный ребенок»). В случае, когда партнер настроен на конструктивный деловой диалог, он занимает позицию «взрослого».

Особенность традиционных компьютерных систем обучения состоит в том, что они жестко моделируют только соподчиненное, возникающее при неравенстве партнеров общение: родитель—ребенок. Обучающая система как бы запрограммирована на роль «родителя». Это выражается, в частности, в том, что диалог инициируется лишь компьютером, его инструкции носят обязательный характер, требуют однозначного подчинения. Обучаемому пользователю отводится роль «послушного ребенка», четко и последовательно выполняющего указания «доминирующего родителя». Однако даже в случае, когда общение ученика с компьютером происходит на равных, оно идет по линии взрослый—взрослый.

В реальной жизни человек, жестко занимающий позицию взрослого, не является компетентным в общении. В. Сатир называет такого человека «компьютер» [11]. Он корректен, но мысли его

\* Первоначальный положительный опыт по использованию среды Лого для психотерапевтической работы с семьей был получен нами в ряде московских школ (№ 1034 и 1126) в ходе совместной работы с Центром психического здоровья (лаборатория кандидата педагогических наук С. Н. Ениколопова) и Институтом новых технологий образования.

чаще всего слишком сложны для понимания, так как они практически лишены привычной и необходимой для восприятия эмоциональной окраски, которую привносит умеренное и умелое использование ролей родителя и ребенка. И хотя в диалоге позиция взрослого, безусловно, является оптимальной для обоих партнеров, она должна являться средством общения, а не довлеющей самоцелью, загоняющей человека в жесткие рамки поведения. Необходимо уметь хорошо играть все роли и свободно их менять по мере необходимости. Ребенок, главным учителем которого является компьютер, подвержен риску стать таким «человеком-компьютером» сам, так как он просто оказывается неспособным воспринимать партнера по диалогу, находящегося в иной роли. В идеале хорошая обучающая программа должна предусматривать реализацию внутри себя всех позиций (родителя, ребенка, взрослого), способствовать выработке у учащегося умения занять какую-то иную позицию самому, а также произвольно менять их, разрешать возникшие в ходе общения ролевые конфликты (играть в игры).

Рассматривая среду Лого с этой точки зрения, отметим, что основная роль в ней принадлежит учащемуся: ему предоставляется возможность в любое время выйти из диалога, инициировать его и не всегда следовать рекомендательным указаниям компьютера (ребенок может иногда не слушаться). С другой стороны, введение образа черепашки позволяет реализовать соотношенное общение (ситуация равенства партнеров). Одна из

основных проблем, решаемых в настоящее время разработчиками среды Лого, — моделирование «черепаших» эмоций: как реализовать непослушание черепашки, отказ выполнить команды ученика, потому что она заболела или устала. А это как раз и есть роль «ребенка».

Таким образом, в результате работы с Лого маленький человек сталкивается с необходимостью принимать решения самостоятельно, причем не только за себя, но и за другого, и нести ответственность за эти решения. Он не только приобретает определенную сумму знаний, но и повышает свою компетентность в более глубоком, личностном плане.

В заключение хочется отметить как психологическую удачу выбор основного персонажа Лого — маленькой черепашки, не претендующей на то, чтобы быть «человеченной». Машина с человеческим поведением вызывает двойственные чувства. С одной стороны, она выступает как воплощение вековой мечты человека, описанной в многочисленных фантастических романах, а с другой стороны, пугает его, обостряет ощущение собственной неполноценности. Достаточно подробно такие амбивалентные чувства ребенка описаны в книге «Приключения Электроника». Главный герой повести живой мальчик Сережа Сыроежкин искренне любит своего электронного двойника и одновременно завидует ему, восхищается им и в то же время винит во всех своих злоключениях, гордится своей исключительностью, связанной с наличием такого уникального друга, и остро переживает ее.

## Литература

1. Берн Э. Игры, в которые играют люди. Л., 1992.
2. Бэндлер Р., Грайндер Д., Сатир В. Изменения совместно с семьями: Пер. с англ. 1976.
3. Велтистов Е. С. Электроник — мальчик из чемодана. М., 1986.
4. Gorbunov A. L. Ecohouse — bildings and energy-efficient home with Lego TC logo//The Computing Teacher. March, 1995. P. 26—29.
5. Лейтц Г. Психодрама: теория и практика. М., 1994.
6. Машбиц Е. И., Ангриевская В. В., Комиссарова Е. Ю. Диалог в обучающей системе. Киев, 1989.
7. Носов Н. А. Психология виртуальной реальности. М., 1994.
8. Пейперт С. Переворот в сознании: Дети, компьютеры и плодотворные идеи. М., 1989.
9. Переверзев Л. Б. Полюбить машины, помогающие учиться//Информатика и образование. 1995. № 5.
10. Петренко В. Ф. К проблеме понимания смысла текста//Интерактивные системы и их практическое применение. Киев, 1984.
11. Сатир В. Как строить себя и свою семью. М., 1992.
12. Соколова Е. Т. Проективные методы исследования личности. М., 1980.
13. Романова, Потемкина О. Ф., Хоме-натаскас Г. Т. Методика «Рисунок семьи»//Общая психодиагностика. М., 1987. С. 206—209.

**А. Л. Семенов,**

*доктор физико-математических наук, ректор МИПКРО,  
заместитель председателя Московского департамента образования*

## МАТЕМАТИЧЕСКАЯ ИНФОРМАТИКА В ШКОЛЕ

В статье А. Л. Семенова «Образование, информатика, компьютеры», помещенной в настоящем выпуске журнала, речь идет об общей стратегии московского образования в отношении школьной информатики. Данная статья посвящена более специальному вопросу, вынесенному в заголовок. Для понимания некоторых ее положений, вероятно, полезно знакомство с предыдущей статьей.

### Предмет математической информатики

Под словом «информатика» понимается широкая область, включающая протягивание трансконтинентальных оптоволоконных кабелей, социально-философские теории, вопросы разработки новых видов полупроводников и т. д.

Фундаментальная естественно-научная часть информатики строит теоретические модели процессов обработки, хранения, передачи информации. По своим объектам, понятиям и методам — это область математики. Предметом ее изучения служат конечные (конструктивные) объекты и алгоритмически описанные (конструктивные) процессы, происходящие в среде этих объектов. Далее эта область будет называться математической информатикой (МИ).

### Соотношение между МИ и информационными технологиями (вычислительной техникой)

С практической точки зрения сегодняшнего дня кажется, что важнейшими устройствами хранения, передачи и переработки информации являются компьютеры. Конечно, это не так. Важнейшими устройствами такого рода являемся мы с вами. Именно попытка механизировать те или иные интеллектуальные действия человека и привела к появлению компьютеров, а попытка построить формальные модели этой деятельности — к созданию МИ. Математической информатике удалось дать точные математические определения того, что такое алгоритм, доказательство, случайность.

МИ появилась и сформировалась в

работах математиков конца XIX — первой трети XX в., огромный вклад в которую внес один из величайших математиков XX в. — Гильберт. Основополагающие результаты МИ были получены перед началом второй мировой войны в работах Геделя, Тьюринга, Поста. С конца 40-х гг. началось бурное развитие вычислительной техники, сразу же востребуя работы теоретиков. Во второй половине XX в. развитие МИ продолжалось, были получены такие фундаментальные результаты, как, например, понятие колмогоровской сложности конечного объекта и выросшая на его базе алгоритмическая теория информации, понятие NP-полной задачи и вся теория сложности вычислений. Однако эти понятия и теории базировались на результатах начального периода и их суммарный вес в действительно фундаментальных вопросах меньше. Процесс открытия важных для вычислительной практики понятий и конструкций в математике продолжался.

Упомянем один из наиболее ярких исторических примеров. Конструкции структурного программирования были введены и первые индуктивные доказательства правильности программ (с использованием инвариантов) были получены А. А. Марковым в конце 40-х гг. и вошли в вычислительную практику в 70-е гг. (Подробный обзор истории МИ и ее соотношения с вычислительной практикой дан в [1, 2]. Развитие информатики, ориентированной в первую очередь на анализ человеческого мышления и языка, тоже продолжалось. Методы математической информатики позволили получить важные для лингвистики результаты.

Итак, фундамент МИ был заложен в попытках моделировать процессы алгоритмической деятельности человека. Ка-



кова же роль компьютеров в современном развитии МИ? Она такая же, как и роль других технологий в развитии соответствующих фундаментальных областей — физики, химии. Компьютерные технологии выступают в качестве источника проблем, формирующего интересы исследователей. Конечно, не прекратились и работы, в которых строятся математические модели мыслительной и коммуникативной деятельности человека. Главное отличие ситуации с МИ от ситуации с физикой и химией заключается в ничтожно малой роли эксперимента. И это не случайно. В физике мы пытаемся объяснить поведение природы, в компьютерной науке мы пытаемся построить нечто новое с заданными свойствами и доказать, что результат достигнут внутриматематическими методами, не прибегая к эксперименту. (Конечно, ситуация здесь соответствует положению в математике в целом.)

### МИ в школьном образовании

Образование — вещь консервативная. В случае математического образования это особенно заметно. Основное содержание курса геометрии не изменилось за 2000 лет. Возможно, однако, что МИ должна быть первым за последние столетия серьезным компонентом, который должен войти в школьное математическое образование. (Пожалуй, только теория вероятностей может еще пожаловаться на несоответствие между важностью концепции вероятности для жизни человека и его отсутствием в школе.) Оправдывает необходимость МИ в школе не бурное развитие вычислительной техники, а действительно фундаментальные открытия МИ, важные для понимания природы человеческого мышления и языка, пределов познания и применимости математики, приложения к различным видам человеческой деятельности. В нашей стране это ясно понимал А. П. Ершов, математик по образованию, информатик (в том числе математический) по профессиональной деятельности, организатор массового введения информатики в советскую школу начиная с 1984 г. [3].

Бурный расцвет новых информационных технологий сыграл решающую роль в фактическом развитии событий. В школьном образовании ряда стран мира появился предмет «Информатика». До-

статочно часто в нем, наряду с разными сведениями (например, что на компьютерную дискету нельзя лить чай и что к ней нельзя подносить магнит), сообщались и некоторые знания из области МИ. Так обстоит дело в России и сейчас. О московском подходе к школьной информатике рассказано в [4].

Отметим, что МИ включается сегодня в систему из трех направлений:

- математическая информатика (иногда называемая еще математикой конечного);
- практические навыки работы со средствами новых информационных технологий, в частности умение писать программы на каком-либо распространенном универсальном языке программирования для компьютера;
- информационная культура, т. е. общее представление об информационных процессах в окружающем мире, об источниках той или иной информации, средствах массовой информации, системе морально-этических и юридических норм, ценностная ориентация.

Министерство образования поместило математику и информатику в одну область знания. Разумеется, можно в этой же области преподавать и нематематические вопросы информатики (например, если с ними хорошо знаком преподаватель математики). Но можно поступать и иначе. Скажем нематематические разделы информатики поместить в образовательную область технологии, что и сделано в московском базовом учебном плане [5].

### МИ и компьютеры в школе

Как мы уже сказали, компьютеры играют меньшую роль в МИ, чем физические экспериментальные установки в физике. К тому же, как правило, эксперимент можно провести, обладая вычислительным устройством, которое всегда под рукой, — собственной головой и руками.

Достаточно важно, чтобы применение компьютера при изучении МИ, так же как и в других дисциплинах, не требовало слишком больших усилий по борьбе с устаревшей технологией. Общепринятый сегодня графический интерфейс, реализуемый на всех современных компьютерах (удовлетворяющих

требованиям Минобразования), позволяет избежать этой борьбы, но необходимо, чтобы конкретное программное обеспечение для МИ использовало бы возможности этого интерфейса и современного оборудования максимально эффективно.

Все-таки, если необходимо использование компьютера, то как это сделать? Первое — в рамках образовательной области в школах без углубленного изучения математики и информатики не пытаться учить детей взрослым языкам программирования. Второе — визуализация (наглядность) — мощное средство в изучении математики. В частности, в случае МИ оно существенно помогает освоить конструкции структурного программирования (робот в лабиринте), переменные и рекурсию (черепашка на плоскости). Учащийся действительно видит процесс выполнения своей программы, и это иногда здорово помогает. В некоторых системах попутно с усвоением упомянутых выше понятий учащиеся могут решать интересные для них задачи (рисовать узоры в Лого). Дополнительная мотивация может делать чудеса в любом школьном предмете.

### **Содержание разделов школьной образовательной области «математика—информатика», их место в учебном плане**

Излагаемые в этом разделе взгляды в полном объеме реализуются только в отдельных экспериментах. Автор предлагает их для обсуждения и совместной работы учителям информатики.

Основной задачей начального образования является развитие мыслительных и коммуникативных навыков ребенка. Конкретные знания, умения и навыки, относящиеся к натуральным числам и словам русского языка, конечно, важны сами по себе, но не менее важно и то, в какую более общую систему целей они включены. В ту же систему целей очень естественно включаются и элементы МИ.

Основная задача первого этапа изучения МИ — знакомство учащихся с формальными языками, предназначенными для описания отношений, утверждений, процессов. Разумеется, элементы этих языков в виде обычной математической символики (знаки арифметических операций, скобки, символы переменных,

функций, символы принадлежности, следствия и т. д.) вводятся в обычных математических курсах. МИ требует дополнительно:

- ряд символов и конструкций (например, символ для выражения «существует» или фиксированную конструкцию для описания процесса «делай операцию P, пока не нарушится условие A»);
- аппарата и практики изучения различных языков: естественных языков разных народов и языков самой МИ методами МИ.

В I классе основные базисные объекты информатики — цепочки символов появляются наравне с числами, вводятся отношения «имя — значение», «предписание — процесс выполнения». С самого начала формируется понятие «правил игры», на базе которого затем будет введено формальное и очень важное понятие стратегии в игре.

Основные конструкции языков математики, в том числе рассматриваемых и используемых математической информатикой, осваиваются к концу начальной школы. Это означает, что учащиеся умеют обращаться с алгебраическими и логическими формулами, конструкциями структурного программирования, функциональными уравнениями, располагают примерами универсальных алгоритмических языков.

Одной из важных традиционных целей курса математики основной школы (прежде всего, геометрии) является освоение понятия (полуформального) математического доказательства. Мы говорим об аксиомах и теоремах геометрии. Конечно, полуформальная система присутствует и в алгебре. Когда мы решаем алгебраическое уравнение, мы фактически осуществляем доказательство в некоторой формальной системе. Осознание этих фактов важнее, чем доказательство конкретных теорем или решение конкретных уравнений. Существующие курсы математики для школы, как правило, пытаются достичь в каком-то смысле полноты. Например, в геометрии выстраивается система теорем об углах, окружностях, треугольниках, параллельных прямых, отвечающая некоему классическому (евклидовскому) образцу. В действительности, можно пожертвовать этой целью полноты, но достичь других, более естественных с общей (не внутриматематической) точки зрения. Эти цели:

- возможность самостоятельного открытия истины;
- представление о непрерывающемся развитии общечеловеческой культуры;
- практическое знакомство с формальными математическими доказательствами и общее представление о них.

О последней цели, имеющей непосредственное отношение к МИ, мы уже сказали. Что же касается первых двух, то геометрия (если избавиться от требования полноты) дает прекрасную возможность для их реализации. История отдельных теорем (и тем) геометрии представляет собой увлекательную драму человеческих идей. Ее рассмотрение в школьном курсе может включать и самостоятельное доказательство, и изучение оригинальных текстов (вплоть до перевода фрагментов — перевод специального, особенного математического текста требует небольшого специального словаря).

Эффективные алгоритмы (например, сортировки), как и общие подходы к их построению («разделяй и властвуй»), также естественно включаются в курс математики основной школы.

Важнейшей темой, доступной учащимся V—VIII классов, является понятие сложности конечного объекта, а также базирующееся на нем алгоритмическое понятие случайности.

Наконец, вероятность — тема непосредственно примыкающая к МИ, а при некотором подходе и включаемая в нее, также должна входить в курс основной школы.

Важнейший раздел МИ — «Отрицательные результаты» — указывает примеры задач, которые невозможно решить с помощью компьютера в принципе или требующие астрономических ресурсов и потому нереальные, а также примеры недоказуемых (хотя и истинных) математических утверждений. (Полное рассмотрение такого примера — знаменитой теоремы Геделя о неполноте — автор использовал в своих лекциях для московских учителей, на английском языке его можно найти в [6].) Изучение этого раздела позволит учащимся выделять на практике такие задачи, которые не следует поручать компьютеру. Начальные примеры таких результатов доступны уже учащимся младшей школы [7], однако, особый смысл они приобретают в старших классах.

Основным полем примеров для построений МИ в школе служат челове-

ские языки (родной, иностранные, искусственные), предписания и рассуждения, всевозможные игры (прежде всего, позиционные игры с полной информацией). Очень важно максимальное привлечение физической активности (возможно в рамках занятий по физической культуре). Необходимо приводить максимальное число примеров из окружающей жизни — ориентация по карте, правила поведения в общественном транспорте, работа телефона-автомата и стиральной машины.

Наличие компьютеров и соответствующего программного обеспечения (Лого, «Роботландия») позволяет расширить круг примеров, многие из которых будут возникать в новой образовательной области технологии (всевозможные конструкторы, вплоть до компьютерного Лего).

Серьезные межпредметные связи с предметами языковых и эстетического циклов, физической культурой, технологией требуют большой работы по подготовке педагогов.

Намеченная выше концепция преподавания МИ и ее места в начальной школе развивается в ряде стран мира в течение десятилетий. Заслуживает внимания венгерский, болгарский, французский опыт [8]. Но наиболее близок нам итальянский подход [9].

Фактически элементы МИ входят во все курсы информатики, ведущиеся в начальной школе (по крайней мере в России). Объясняя детям конструкции Лого, «Роботландии», даже Бейсика, учитель вынужден вводить их и в мир МИ. Бывает, как и в других аналогичных случаях, что талантливый учитель делает это с блеском. Однако такое введение страдает отсутствием систематичности и ясности, поскольку ставит прежде всего цели использования компьютера обучения языку программирования.

Систематический интегративный курс «Язык и математика» для начальной школы, включающий МИ, уже в течение 7 лет используется в нескольких школах Москвы, а последние годы — в Чехии и Италии [10].

## Первые практические шаги и перспективы

Наиболее продуктивный подход по введению МИ в начальное образование не может быть реализован сразу. В частности, он требует достаточно серьезной

работы по переподготовке учителей начальной школы. В то же время постепенный сдвиг в этом направлении уже происходит.

Очередной этап, осуществление которого возможно уже сейчас, должен состоять в разработке курса МИ для средних классов, может быть, небольшого объема (скажем, до 20 часов), который используется в качестве дополнения к курсу математики в V—VIII классах. Изучение материала естественно организовать в виде 2—4 модулей. В периоды прохождения модуля в составе интегрированной области «математика—информатика» интенсивность классных занятий — 2—3 часа в неделю. Этот курс должен быть сориентирован на беском-

пьютерное преподавание, хотя, как и в случае других предметов, возможно создание компьютерной поддержки для него на различных моделях компьютеров. Некоторым приближением к такому курсу, правда, большего объема, может служить книга, созданная в рамках советско-американского проекта Института Норриса (Миннесота) и Института новых технологий образования (Москва). Вариант курса для дополнительного образования предлагался в [12]. Сценарии такой поддержки, как и проработки уроков, совмещенных с уроками родного, иностранного языков, технологии, искусства, физической культуры должны войти в отдельное пособие, сопровождающее курс.

### Литература

1. Семенов А. Л., Успенский В. А. Математическая логика в вычислительных науках и вычислительной практике// Вестник Академии наук СССР. 1988. № 7. С. 93—103.

2. Успенский В. А., Семенов А. Л. Теория алгоритмов: основные открытия и приложения. М.: Наука, 1987. — (Б-чка программиста).

3. Ершов А. П., Звенигородский Г. А. Зачем нужно уметь программировать?//Квант. № 9. С. 47—51.

4. Семенов А. Л. Образование, информатика, компьютер//Информатика и образование. № 5. 1995.

5. Московский базовый учебный план//Московский департамент образования. Москва, 1994.

6. *Semenov A. A simple detailed proof for Goedel's incompleteness theorem*//Kibernetika, Academia, Praha, 1988. V. 24, № 6. P. 447—451.

7. Успенский В. А., Семенов А. Л. Решимые и нерешимые алгоритмические проблемы//Квант. № 7. С. 9—15.

8. Язык и математика: Учебное посо-

бие для начальной и средней школы. Проблемная группа по образованию. София, 1984.

9. Guida Alla Formazione Matematica Del Primo Ciclo Elementare. Parte Prima - Terza. Progetto Ricme. Rinnovamento del Curricolo Matematico Elementare, - CETEM, Armando Editore, Milano — Roma, 1985.

10. Математика и язык. Рабочие тетради для начальной школы 1—15: Книги для учителя 1—15: Проекты. Москва, Институт новых технологий образования, 1995.

11. Алгоритмика, комплект учебных материалов для средней школы. Институт новых технологий образования. М., 1991—1995.

12. Простое и сложное в программировании. М.: Наука, 1988. 176 с.: ил. — (Сер. «Кибернетика — неограниченные возможности и возможные ограничения»).

И. В. Гадолина, Т. Д. Константинова, А. Д. Пудиков,  
В. Б. Репеченко, А. А. Семенов, П. А. Юдина,

*преподаватели средней школы № 188, Москва*

## ИСПОЛЬЗОВАНИЕ ЕСТЕСТВЕННО-НАУЧНЫХ ДИСКОВ CD-ROM В ОБРАЗОВАТЕЛЬНОМ ПРОЦЕССЕ

В настоящее время существует большое количество лазерных компакт-дисков энциклопедического характера, имеющих образовательную направленность. Наша школа, получив от Института новых технологий образования новый компьютерный класс, оснащенный машинами Macintosh, а также проекционным аппаратом «Проксима», заинтересована в наиболее широком их применении, в том числе и для преподавания естественно-научных дисциплин, таких, например, как биология. В ряде московских школ компьютеры Macintosh уже используются для преподавания математики и физики (программы «Живая математика» и «Живая физика»), а также для проведения уроков английского языка («Живые книжки»). Применение же естественно-научных дисков пока недостаточно отработано. Среди причин того, что великолепно выполненные разнообразные диски не вошли еще в практику преподавания, не последнюю роль, вероятно, играет то, что они написаны на английском языке. Среди авторов статьи — учителя информатики, английского языка и биологии. Подробно ознакомившись с диском «Киты и дельфины» (выбор диска был сделан достаточно случайно), мы задумались над вопросом: каким образом можно использовать этот материал в учебном процессе?

То, что на первый взгляд представлялось недостатком, а именно отсутствие русифицированного варианта программы, особенно заинтересовало учителей английского языка. Им показалось полезным использовать печатные и звуковые экраны диска для отработки навыков перевода.

Перед учениками английской группы была поставлена цель — подготовить учебный материал для проведения уроков биологии в VI—VII классах. Дополнительными побудительными стимулами к работе над переводом должны были стать красочный экран, на котором мож-

но было просмотреть небольшие документальные фильмы, и звуковое сопровождение — голоса китов.

При проведении уроков английского языка с использованием диска ученики работали, как обычно, со словарями, предварительно получив задание ознакомиться с данной тематикой по энциклопедии «Жизнь животных» (или по другим источникам). Это способствовало пониманию необходимости владения научным материалом, хотя бы в рамках энциклопедии, для выполнения качественного научного перевода. Для учащихся X—XI классов одного экрана (примерно 600 печатных знаков) было достаточно на один двоянный урок. Те ученики, которые не справлялись с заданием, получали распечатки текстов на дом.

Опыт показал, что один и тот же материал следует давать независимо двум-трем ученикам с целью последующей проверки и обсуждения. При такой работе по сопоставлению переводов ученики знакомятся с понятиями: стилистическая ошибка, неточность, неверный смысл. Обсуждаются проблемы литературного стиля.

Не секрет, что лингафонные кабинеты почти везде прекратили свое существование, поэтому наиболее полезная и необычная для детей часть английской программы — это работа со звуковыми экранами. Текст, произносимый диктором, никогда не совпадает с тем, что написано на экране (по-видимому, это сделано для большего охвата информации). Он начитывается в нормальном темпе и с первого раза с трудом воспринимается на слух даже преподавателями. Большое удобство для детей — это возможность повторного и многократного прослушивания, причем с любого места, — в этом, кстати, тоже существенное преимущество использования дисков для работы с текстом на слух, так как на магнитофоне очень трудно осуществить повторный запуск с нужного места. Рукописные пере-

воды подготавливаются учениками с использованием текстовых процессоров в виде печатных материалов, затем переплетаются и, снабженные дополнительным материалом для поиска нужного текста, передаются вместе с инструкцией учителю биологии. В инструкции, помимо прочих необходимых материалов, содержится рисунок экрана с указанием полей (на рисунке показан пример экрана с пояснениями), что облегчает работу пользователя, не привыкшего к подобным программам.

Таким образом, представляется возможным проведение серии английских уроков, объединенных общей тематикой и общей (в рамках класса) целью в объеме примерно 10 часов.

Следующим этапом этой работы является проведение уроков биологии в VII классе и естествознания в VI классе (последнее проходило в нашей школе в

рамках эксперимента). Здесь занятие начинается с показа программы на экране. При проведении урока учитель может не только пользоваться текстовыми материалами, предоставляемыми программой, но и дополнять или заменять материал, используя лишь изобразительную часть программы и голоса китов.

На основании материала диска могут быть проведены уроки: «Экосистема, взаимосвязи видов», «Явления в популяции: популяция и ее состав» и «Явления в экосистеме: круговорот веществ». Система вложенных экранов диска (один экран — вид, из него осуществляется вызов экранов подвидов) позволяет ученикам исключать путаницу в понятиях «вид» и «подвид», а самостоятельная работа с программой во второй части урока способствует включению в учебный процесс активной памяти и позволяет ученикам понять, что такое популяция.

## П. Гангнус

### ПРАВДА ОБ ОШИБКЕ ПЕНТИУМА

Каждому из нас приходится хотя бы раз в жизни работать с какой-нибудь таблицей, например школьными таблицами Брадиса. Один взгляд на пересечение нужных столбца и строки избавляет от долгих вычислений. Когда приходится заботиться о времени, программы тоже используют таблицы с заранее вычисленными результатами. Пока таблица верна, все хорошо.

При создании Пентиума одной из целей было увеличение быстродействия операций с плавающей точкой в три раза по сравнению с 486-м процессором. Для этого инженеры INTEL должны были улучшить традиционный алгоритм деления «сдвиги-и-вычит», который за такт дает лишь один значащий бит. Был использован новый метод, называемый по первым буквам имен авторов алгоритмом SRT, который дает два бита за такт.

Для вычисления промежуточных цифр SRT использует справочную таблицу, которая была включена в Пентиум. 2048 клеток, из них 1066 содержат +2 или -2, а все остальные — нули. В качестве указателя в таблице алгоритм использует двоичное разложение делителя.

К сожалению, инженер, приготовивший и загрузивший таблицу, ошибся, и пять значащих биты загружены неверно. Вдобавок таблицу никто не проверял. Эти пять значащих расположены вдоль края таблицы и должны содержать +2. А там — 0.

Так как алгоритм SRT рекурсивный, это может практически никак не отразиться на результате. А может дать ошибку в четвертой значащей цифре. Вероятность этого — около 1/360 млрд. Чаще — в 1/5 млрд. случаев — ошибка случается в 9–10-м знаке.

Таким образом, ошибка Пентиума — ошибка не в металле, а в программе — такую любой программист может понять. Но не оправдать!

Спорят о том, насколько важна эта ошибка. INTEL уверяет, что обычный пользователь электронных таблиц столкнется с ней раз в 27 тыс. лет. IBM, которая в декабре выкинула пентиумные системы со своих складов, считает, что это случается каждые 24 дня. Кто прав?

Полагаю, это несущественно. В любом случае производителем software придется «обходиться» эту ошибку в своих программах. А пока у вас нет таких программ, вы никогда не сможете доверять Пентиуму с ошибкой.

Эта история дает нам еще один урок: каждая программа должна контролировать результаты счета! Мы все это знаем, но часто ли делаем?

**Н. В. Болотова, М. В. Деревягина, Е. Ф. Ржевский,**

*средняя школа № 1745, Москва*

## **КОМПЬЮТЕР НА УРОКАХ ТЕХНОЛОГИИ (первый опыт и ближайшие перспективы)**

В этом году школьной информатике в нашей стране исполнилось 10 лет. Те, кто начинал ее преподавать (особенно вдали от Москвы), помнят безмашинную информатику. Да и на первых компьютерах учили в основном Бейсику или Паскалю, а не приложению этих языков для решения содержательных, значимых задач. Сегодня стало окончательно ясно: программирование как самоцель, без использования его в других учебных предметах, для большинства учащихся малоинтересно и не приводит к заметным успехам. Вместе с тем, по мере того как в школах появляются современные компьютеры и программные продукты, расширяется возможность сочетать информатику с другими школьными дисциплинами. И прежде всего с теми, которые требуют формирования навыков моделирования, конструирования и алгоритмизации. Одним из таких предметов является технология.

### **Исходные позиции**

Располагая классом Macintosh, мы попытались «интегрировать» информатику с технологией на основе разработанной нами экспериментальной программы при поддержке Института новых технологий образования. Курс технологии у нас начинается с освоения простейших операций по обработке бумаги и дерева; далее идет изготовление отдельных заготовок, деталей и модульных блоков; затем сборка из них различных конструкций (моделей автомобилей, вертолетов и т. д.) с постепенным увеличением сложности их состава и «плотности» материала. Следующим этапом приближения этих моделей к реальности является переход к металлу — классическому детскому «конструктору». Однако примитивный набор его исходных элементов и ограниченное число их комбинаций мало расширяет круг модельно-конструктивных задач. Несравненно больший диапазон учебно-технологических возможностей поддерживают наборы «конструкторов» фирмы Lego-Dacta. Они позволяют не только собирать множество моделей весьма сложных механизмов и машин, но и управлять ими через компьютерную среду Control Lab по программе, созданной самими учащи-

мися на языке Лого. Таким образом, открывается реальная перспектива интегрированного курса «Информатика—технология» при согласовании целей, содержания и методик, которых придерживаются учителя этих дисциплин.

Вот как это происходило у нас в школе.

### **Эксперимент**

Получив от Института новых технологий наборы Lego-Dacta: 9701 и 9702, наша группа приступила весной 1995 г. к апробации программы по обучению школьников V — IX классов основным технологическим принципам проектно-конструкторских работ. В плане информатики необходимо было, изучить с одной стороны, воздействие компьютерных команд на поведение моторов, света и звукоизлучателей, а с другой — влияние обратной связи от выходных параметров на входные через датчики угла поворота и фоторезисторы. Для участия в эксперименте было отобрано восемь учащихся VII и VIII классов, наиболее подготовленных к модульному конструированию.

Всего состоялось 8 занятий по 2 урока, каждое по следующей тематике:

1. Уточнение конструктивных особенностей модульной системы Lego и возможностей ее использования при создании моделей. Определение назначения и способов применения двигателей, углового и светового датчиков. Конструирование простейших движущихся моделей с двумя типами передач: шкивной и зубчатой.

2. Изучение первых команд: tto "motora, on, off, onfor, rd, wait, repeat, setpower, setleft, setright и подключение моторов к интерфейсу.

3. Сборка модели с двумя моторами. Управление движением модели: передний и задний ход, поворот за счет остановки одного мотора.

4. Составление программы движения модели по некоторой заданной траектории.

5. Понятие обратной связи. Знакомство с угловым датчиком (angle#). Освоение команд: resetrotation, waituntil.

6. Знакомство со световым датчиком (light#). Конструирование модели кон-

вейера, реагирующего на детали определенного цвета (например, желтого).

7. Конструирование системы из нескольких различных моделей-блоков. Конвейер, «Рука—сжать», Поворот, «Рука—расжать», Обратный поворот, Лифт (подъем—опускание).

8. Заключительное оформление и документирование проекта с сохранением всех процедур.

### Предварительные выводы

Комплект Lego-Dacta со средой Control Lab вызывает огромный интерес у школьников V—IX классов. Однако для получения максимального учебно-познавательного эффекта целесообразно начинать знакомство с идеей модульности и компьютерного управления уже в начальной школе. Создавая на этой базе действующие модели различных машин и систем, школьники гораздо глубже усваивают такие понятия, как конструктивные узлы и соединения, виды движения и способы его передачи, датчики и сигналы, исполнительные механизмы и составление управляющих ими программ. Работа с компьютером пробуждает у детей интерес к рациональному проектированию технических объектов и заметно сокращает сроки перехода от их замысла (эскиза, схемы, чертежа) к материальному воплощению. В итоге информатика становится для школьников действенным инструментом решения инженерно-конструкторских, дизайнерских и других технологических проблем. Последние же, в свою очередь, мотивируют интерес к углубленному изучению языка и процедур программирования. Наиболее полная интеграция информатики и технологии в едином учебном процессе требует, чтобы занятия (во всяком случае на начальных этапах) проводились совместно преподавателями двух названных дисциплин.

### Дальнейшие планы

На основании полученных результатов намечена экспериментальная программа курса «Информатизированная технология» для учащихся V—IX классов на 1995/96 учебный год. Она охватывает весь цикл технологического обучения: формулирование первоначальной идеи, проектно-конструкторскую разработку и воплощение в материале конечного продукта. Ее главная цель — сочетать наглядно-практическое освоение предмета с теоретической рефлексией всех производимых при этом операций и действий. Выполнение второго условия осуществляется благодаря привлечению концептуального аппарата информатики. Тем самым обеспечивается единство системы

дидактических методов, средств и приемов, делающих возможным плавный переход от решения элементарных задач к созданию моделей промышленных роботов.

Основные этапы программы:

#### 1. Элементарное конструирование.

Понятие о модуле и модульном построении конструкций. Степени сложности.

Знакомство с набором Lego; объемное конструирование из элементов простейших неподвижных объектов.

Понятие о движении и способах его передачи. Сборка простейших кинематических узлов.

#### 2. Конструирование механизмов с электродвигателем.

Знакомство с электродвигателем, источниками (батареями) питания, электроцепями, выключателями и переключателями.

Создание самодвижущихся тележек на одном моторе, с прямым и обратным ходом на шкивной и зубчатой передачах. Передаточное число.

Создание усложненных самодвижущихся тележек на двух моторах со шкивными, зубчатыми и червячными передачами, с поворотом в стороны (за счет остановки одного мотора) при использовании датчика поворота. Понятие сигнала и циркулирования информации в контуре управления.

#### 3. Системы компьютерного управления, команды и составление программ.

Отработка движения тележек по осевой линии с выполнением команд «влево—вправо», «влево—стоп—вправо» и т. д.

Составление и отладка программы движения по плоскости с командами «вперед—назад», «вперед—стоп—назад» и т. д.

Составление и отладка программы сигнала «стоп» — «опасно» с применением светового датчика.

Составление и отладка программы поворотов с применением углового датчика.

#### 4. Основы робототехники. Проектирование и конструирование.

Сборка комплексов из нескольких передач-редукторов (шкивных, зубчатых, червячных, гребенчатых), отдельно переключаемых и контролируемых датчиками.

Создание на основе комплексов передач моделей единичных роботов.

Составление и отладка программ компьютерного управления системами из единичных роботов и многокомпонентных исполнительных механизмов.

Таковы первые шаги, сделанные нами в направлении интегрированного курса «Информатика—технология».



**В. М. Нестеров,**

*кандидат физико-математических наук,*

*руководитель отдела информатизации Института новых технологий, Санкт-Петербург*

## **КОМПЬЮТЕРНЫЙ МИНИ-КЛАСС ДЛЯ СЛЕПЫХ И СЛАБОВИДЯЩИХ ДЕТЕЙ**

Статья кратко суммирует опыт, приобретенный сотрудниками Санкт-Петербургского филиала Института новых технологий при составлении проекта оборудования компьютерного класса для использования в специальных школах-интернатах для слепых и слабовидящих детей. Написанию статьи предшествовал анализ опыта работы школы-интерната № 1 для слепых и слабовидящих детей Санкт-Петербурга и школы-интерната № 1 Москвы. Также рассматривалась книга «Independence day (designing computer solutions for individuals with disability)», Apple Computer, 1990. Одним из основных источников данных по конкретным программным и аппаратным устройствам служила электронная база данных, поставляемая на CD-ROM «CO-NET (Cooperative Database Distribution Network for Assistive Technology)», 7-th edition, 1994, информация, распространяемая по телекоммуникационным каналам организацией International Braille and Technology Center for the Blind (USA), информация общества EASI (Equal Access to Software and Information) и техническая информация фирм-производителей. Всего было проанализировано около 640 технических документов, описывающих конкретные программные и аппаратные средства для оснащения компьютеров, используемых детьми с теми или иными физическими недостатками.

Ниже кратко рассмотрены требования к компьютерам и периферийному оборудованию общего назначения и несколько более подробно — состав специального аппаратного и программного обеспечения такого компьютерного класса.

### **1. Общая конфигурация компьютерного класса**

При изучении отечественного и зарубежного опыта работы со слепыми и слабовидящими детьми было установле-

но, что оптимальной комплектацией является «компьютерный мини-класс с локальной сетью, оснащенный дополнительным специальным оборудованием и подключенный к глобальным телекоммуникационным сетям».

Слепые дети, безусловно, нуждаются во внимании преподавателя больше, чем обычные школьники. Поэтому вместо обычного для общеобразовательных школ компьютерного класса из 10—12 машин в специальных интернатах следует рекомендовать установку небольших компьютерных классов из 4—5 машин, которые должны быть связаны между собой локальной сетью. Локальная сеть позволит рационально использовать как ресурсы общего назначения (память жестких дисков, принтер, CD-ROM), так и специальные ресурсы (брайлевский принтер). Подключение к глобальной телекоммуникационной сети предоставляет дополнительные возможности по обмену информацией с другими российскими и международными компьютерными центрами.

Следует учитывать тот факт, что специальные школы часто уже оборудованы вычислительной техникой, состоящей из одиночных IBM-совместимых компьютеров. Они могут быть подключены к сети устанавливаемого мини-класса, что без дополнительных затрат позволит более эффективно использовать их в учебном процессе.

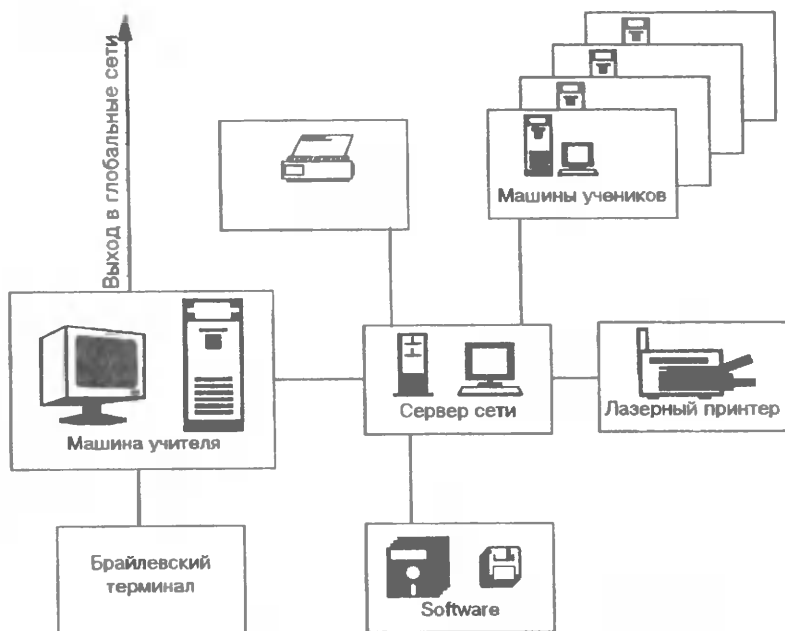
Таким образом, в качестве базового комплекта компьютерного мини-класса следует признать следующий:

- машина учителя;
- 4 машины ученика;
- сервер сети.

### **2. Компьютеры**

Рассмотрим более подробно каждую из входящих в компьютерный класс машин.

Сервер сети является необходимым при использовании современных вер-



Конфигурация компьютерного класса

сий сетевого программного обеспечения NetWare. В качестве сервера наиболее целесообразно использовать машину 486DX-50 или 60 мГц с жестким диском не менее 210 Мб (а лучше 340 Мб). Оперативная память не менее 4 Мб. Требования к монитору минимальные — моно SVGA.

Машина учителя должна использоваться в двух целях. Во-первых, для его работы, во-вторых, для работы учеников в тех случаях, когда требуется специальное оборудование, не установленное на ученических машинах. Машина учителя должна комплектоваться высококачественным цветным монитором, желательно размером не менее 17". Несмотря на то что такие мониторы весьма дороги, включение хотя бы одного позволит использовать специальные программы для слабовидящих детей, которые ориентированы на крупное изображение текста и графических образов на экране. Основные характеристики машины должны быть приблизительно следующими: 486 SX, 33 мГц, 170 Мб, 4 Мб RAM.

В качестве ученических машин следует установить обычные 386-е SX машины со следующими характеристиками: 40 мГц, 2 Мб RAM. Так как они будут в основном использоваться слепыми детьми, их достаточно оборудовать

моно SVGA мониторами или половину машин цветными, половину — черно-белыми мониторами.

### 3. Дополнительное оборудование общего назначения

Для установки сети все компьютеры должны быть оборудованы сетевыми картами Ethernet 16 bit.

Для наиболее эффективного использования компьютерного мини-класса он должен быть укомплектован необходимым периферийным оборудованием. Наличие сети позволит его использовать в равной мере всеми работающими пользователями.

В качестве общего принтера рекомендуется установка лазерного принтера HP LJ 4L 300, который совмещает в себе умеренную цену и большие возможности. Принтер даст возможность как производить обычные распечатки, так и применять специальные программы для людей с ослабленным зрением, которые выводят на печать информацию в сильно увеличенном масштабе.

В последнее время получили большое распространение телекоммуникационные сети. Слепые дети и дети со слабым зрением путем участия в теле-

**ИНТ** - один из мировых лидеров в области новых технологий образования

**ИНТ** ведет работу со школами России, разрабатывая программы развития информационного пространства




**ИНТ** обеспечивает подготовку и методическую поддержку учителей в области использования новых информационных технологий

**ИНТ** предлагает российским школам весь спектр образовательных решений, компьютеры Макинтош, образовательные конструкторы Lego Dacta, презентационные средства Medium, компьютерные естественно-научные лаборатории, видео и издательские системы






## ИНТ - ОБРАЗОВАТЕЛЬНЫЙ ПАРТНЕР Apple В РОССИИ

### ПОЧЕМУ APPLE ВЫБРАЛ ИНТ?

-  - восьмилетний опыт работы с компьютерами Macintosh
-  - тесные связи с российской системой образования
-  - собственные разработки мирового уровня

### ПОЧЕМУ ИНТ ВЫБРАЛ APPLE?

-  - дружественный интерфейс, эргономичность компьютеров
-  - компьютерные технологии завтрашнего дня
-  - поддержка образовательных пользователей



СПРАВКИ ОБ УЧЕБНОЙ МЕТОДИЧЕСКОЙ И ПЕЧАТНОЙ ПРОДУКЦИИ  
СЕМИНАРАХ И ВЫСТАВКАХ

(095) 915-6292  
915-1394

СПРАВКИ ПО ВОПРОСАМ КОМПЬЮТЕРНОЙ ОБРАЗОВАТЕЛЬНОЙ ТЕХНОЛОГИИ И ПРОДУКЦИИ:  
ДЛЯ КОМПЬЮТЕРОВ МАКИНТОШ  
ДЛЯ КОМПЬЮТЕРОВ ИВМ

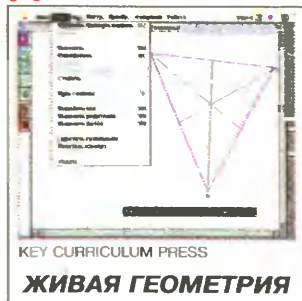
(095) 915-6214  
915-6292

АДРЕС: ИНТ, РОССИЯ, 109004, МОСКВА, Н. РАДИЩЕВСКАЯ, 10  
ФАКС: (095) 915-6963, E-MAIL: INT@INT.GLAS.APC.ORG



ИНСТИТУТ НОВЫХ ТЕХНОЛОГИЙ ОБРАЗОВАНИЯ

## ОБУЧАЮЩИЕ И РАЗВИВАЮЩИЕ СРЕДЫ ДЛЯ РОССИЙСКОЙ ШКОЛЫ



KEY CURRICULUM PRESS

### ЖИВАЯ ГЕОМЕТРИЯ

Электронный альбом для геометрических построений и экспериментов. Наиболее успешный пример программно-педагогического продукта, адаптированного к российской программе и поддерживающего основные школьные учебники. Очень эффективен для работы с учащимися всех уровней - от "слабых" до "олимпиадчиков".



KNOWLEDGE REVOLUTION

### ЖИВАЯ ФИЗИКА

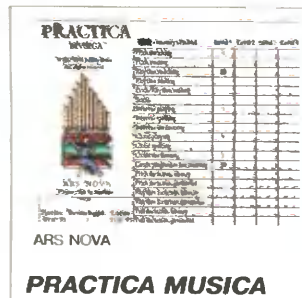
Высокоинтерактивная проектная среда для изучения основных разделов школьной физики. Учитель и ученики, рисуя мышкой на экране, могут сами создавать задачи, эксперименты, демонстрации, наблюдать за взаимодействиями созданных объектов, отображать состояние моделируемой системы графиками и векторами. Методическое сопровождение программы содержит несколько десятков готовых физических задач и моделей экспериментальных установок.



LCSI

### ЛОГОМИРЫ

Универсальная образовательная среда на базе языка ЛОГО, развивающая интеллект, познавательные способности и творческое мышление детей. Среда для создания мультимедиа детьми и учителями, адекватная современной технологии и философии образования, с помощью которой малыши увлеченно учатся рисовать, писать и читать, оживлять и озвучивать забавные картинки; старшеклассники - строить модели природных и социальных явлений; учителя - разрабатывать собственные программы и обучающие модули.



ARS NOVA

### PRACTICA MUSICA

Обучающая музыкальная программа. Учащемуся предлагается вначале ознакомиться с теорией, изложенной в сопровождающей программу книге, а потом заняться практической работой - тренировкой музыкального слуха, умения распознавать ноты, интервалы и аккорды. Программа содержит сборник музыкальных диктантов из восточной библиотеки мелодий. Каждый из девятнадцати разделов программы включает в себя четыре уровня сложности.



ИНТ

### НАША ВСЕЛЕННАЯ

Содержание программы охватывает наиболее важные аспекты устройства Солнечной системы и направлено на углубленное понимание и расширение наглядно-образных представлений детей о движении планет Солнечной системы, работе искусственных спутников Земли, наблюдаемых с Земли космических явлениях. Программа предназначена для использования на уроках естествознания в начальных классах, в курсе астрономии средней школы.



LEGO DACTA

### КОМПЬЮТЕРНОЕ ЛЕГО

Лего - самый популярный в мире конструктор. Ребенок может собрать свою собственную модель, заставить ее передвигаться, анализировать окружающую среду с помощью специальных датчиков и реагировать на внешние воздействия. Для этого нужно подсоединить модель к компьютеру и написать программу в среде Лого. В комплект для российских школ входят методические материалы по новой области образования - Технологии.

СПРАВКИ ОБ УЧЕБНОЙ МЕТОДИЧЕСКОЙ И ПЕЧАТНОЙ ПРОДУКЦИИ  
СЕМИНАРАХ И ВЫСТАВКАХ

(095) 915-6292  
915-1394

СПРАВКИ ПО ВОПРОСАМ КОМПЬЮТЕРНОЙ ОБРАЗОВАТЕЛЬНОЙ ТЕХНОЛОГИИ И ПРОДУКЦИИ:  
ДЛЯ КОМПЬЮТЕРОВ МАКИНТОШ  
ДЛЯ КОМПЬЮТЕРОВ IBM

(095) 915-6214  
915-6292

ЦРЭС: ИНТ, РОССИЯ, 109004, МОСКВА, Н. РАДИЩЕВСКАЯ, 10  
АКК: (095) 915-6963, E-MAIL: INT@INT.GLAS.APC.ORG

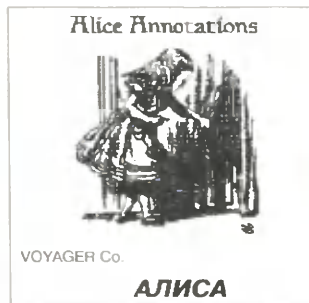


## РАЗВИВАЮЩИЕ ПРОГРАММЫ, СРЕДСТВА И ИГРЫ ДЛЯ ШКОЛЫ И ДОМА

РАЗВИТИЕ ИНТЕЛЛЕКТА, А НЕ ЗУБРЕЖКА



Интерактивные компакт-диски обеспечивают погружение в среду иностранного языка. Позволяют детям самим читать сказки или попросить компьютер прочитать их вслух. Методические пособия на русском языке.



Voyager - ведущее издательство электронных книг, выпускающее десятки классических произведений английской и американской литературы с подробными комментариями и возможностью самостоятельной работы учителей и учащихся. Для специализированных школ с изучением английского языка.



Программа, позволяющая создавать тексты, рисунки, коллажи и т.п. Содержит более 80 готовых фигурок, букв, которые малыши могут использовать в своих работах (каждую букву и цифру компьютер называет вслух); а также различные звуки и специальные эффекты для работы с экраном.

## Sim - знаменитая серия моделирующих образовательных игр от фирмы Maxis



**ОБУЧЕНИЕ  
ПРЕДПРИНИМАТЕЛЬСТВУ  
И БИЗНЕСУ ЧЕРЕЗ ИГРУ**



Программа представляет уникальную возможность взглянуть изнутри на жизнь муравьиного сообщества и, используя различные ролевые ситуации, исследовать взаимоотношения этих интересных общественных насекомых между собой и с окружающей природой.



Биологическая программа, позволяющая проводить эксперименты по генной инженерии, моделированию окружающей среды и экосистем, исследованию эволюционных процессов и даже изменению физических основ воображаемой Вселенной.

СПРАВКИ ОБ УЧЕБНОЙ МЕТОДИЧЕСКОЙ И ПЕЧАТНОЙ ПРОДУКЦИИ  
СЕМИНАРАХ И ВЫСТАВКАХ

(095) 915-6292  
915-1394

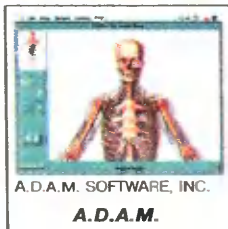
СПРАВКИ ПО ВОПРОСАМ КОМПЬЮТЕРНОЙ ОБРАЗОВАТЕЛЬНОЙ ТЕХНОЛОГИИ И ПРОДУКЦИИ:  
ДЛЯ КОМПЬЮТЕРОВ МАКИНТОШ  
ДЛЯ КОМПЬЮТЕРОВ ИВМ

(095) 915-6214  
915-6292

АДРЕС: ИНТ, РОССИЯ, 109004, МОСКВА, Н. РАДИЩЕВСКАЯ, 10  
ФАКС: (095) 915-6963, E-MAIL: INT@INT.GLAS.APC.ORG

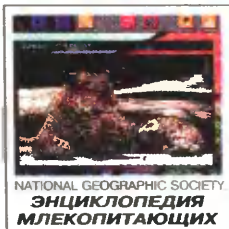


## СЛОВАРИ И ЭНЦИКЛОПЕДИИ НА CD-ROMax



A. D. A. M. SOFTWARE, INC.  
**A. D. A. M.**

Богатейший анатомический справочник для средних школ. Все системы и органы человека.



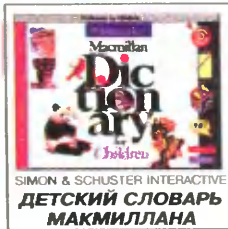
NATIONAL GEOGRAPHIC SOCIETY  
**ЭНЦИКЛОПЕДИЯ МЛЕКОПИТАЮЩИХ**

Энциклопедия более чем 200 видов млекопитающих из разных уголков мира. Содержит более 700 страниц текста, более 700 фотографий, 140 записей голосов животных, карты с изображением их ареалов обитания, 28 видеофрагментов о жизни различных зверей в природе, а также увлекательную игру.



REMEDIA, INC.  
**БАБОЧКИ МИРА**

Энциклопедия, содержащая информацию о разнообразии, экологии, жизненном цикле и анатомии более 250 видов бабочек. Цветные иллюстрации и текстовые описания рассказывают о каждом из семейств, от крупных и ярких тропических видов до самых обычных бабочек.



SIMON & SCHUSTER INTERACTIVE  
**ДЕТСКИЙ СЛОВАРЬ МАКМИЛЛАНА**

Самый популярный компьютерный словарь для детей, начинающих изучать английский язык. Мгновенный поиск слова, даже написанного с ошибкой; увлекательные языковые игры.



SOFTWARE TOOLWORKS, INC.  
**АТЛАС МИРА**

Атлас мира содержит карты стран, планы некоторых городов, возможности поиска и навигации, разнообразную статистическую информацию и средства ее представления (таблицы, диаграммы и т.п.)



MICROSOFT CORP.  
**МУЗЫКАЛЬНЫЕ ИНСТРУМЕНТЫ**

Интерактивная энциклопедия о более чем 200 музыкальных инструментах; с возможностью поиска по алфавиту, по группам музыкальных инструментов и по месту происхождения. Можно увидеть и услышать каждый инструмент.



COMPTON'S NEW MEDIA  
**ЭНЦИКЛОПЕДИЯ КОМПТОНА**

Интерактивная энциклопедия, содержащая разные сведения по многим отраслям знаний. Информация представлена во всех возможных видах (текст, видео, фото, звуки, музыка).



AHC  
**КАТАЛОГ МУЗЕЯ ГЕНРИ ФРИКА**

В каталоге содержатся изображения 241 шедевра западноевропейской живописи из коллекции американского предпринимателя и знатока живописи Генри Фрика.

## УЧЕБНИКИ И МЕТОДИЧЕСКИЕ ПОСОБИЯ С КОМПЬЮТЕРНОЙ ПОДДЕРЖКОЙ

"Алгоритмика" - курс развития алгоритмического мышления для 6-8 классов - создан в ИНТе и рекомендован Министерством образования РФ для российских школ (пособия для учителя и ученика, задачник).

"Язык и математика" - курс мышления и коммуникации для начальной школы.  
"Конструкторы Lego-Logo. Сборник проектов", "Геометрия в лицах", и другие методические пособия.

### КРОМЕ ТОГО

Тесты. Тренажеры. Редакторы. Корректоры. Программа для связи - First Class. Электронные таблицы. Утилиты. А также сотни программ для МАК и IBM.

СПРАВКИ ОБ УЧЕБНОЙ МЕТОДИЧЕСКОЙ И ПЕЧАТНОЙ ПРОДУКЦИИ СЕМИНАРАХ И ВЫСТАВКАХ

(095) 915-6292  
915-1394

СПРАВКИ ПО ВОПРОСАМ КОМПЬЮТЕРНОЙ ОБРАЗОВАТЕЛЬНОЙ ТЕХНОЛОГИИ И ПРОДУКЦИИ: ДЛЯ КОМПЬЮТЕРОВ МАКИНТОШ ДЛЯ КОМПЬЮТЕРОВ IBM

(095) 915-6214  
915-6292

АДРЕС: ИНТ, РОССИЯ, 109004, МОСКВА, Н. РАДИЩЕВСКАЯ, 10  
ФАКС: (095) 915-6963, E-MAIL: INT@INT.GLAS.APC.ORG

Дизайн Domino

коммуникационных проектах смогут на равных общаться со сверстниками, имеющими нормальное зрение. Для участия в телекоммуникационных проектах, которые для рассматриваемого контингента учащихся служат настоящим «окном в мир», компьютерный класс необходимо оборудовать модемом со скоростью передачи данных не менее 9600 bps и с возможностью поддерживать современные протоколы передачи данных, равно как с возможностью посылать и принимать факсовые сообщения.

Весьма желательной является включение в состав класса сканера, например HP ScanJet 2P.

В последнее время программное обеспечение и базы данных часто распространяются на лазерных дисках. В дальнейшем эта тенденция будет усиливаться, и оборудование класса дисководом для чтения лазерных дисков является совершенно необходимым.

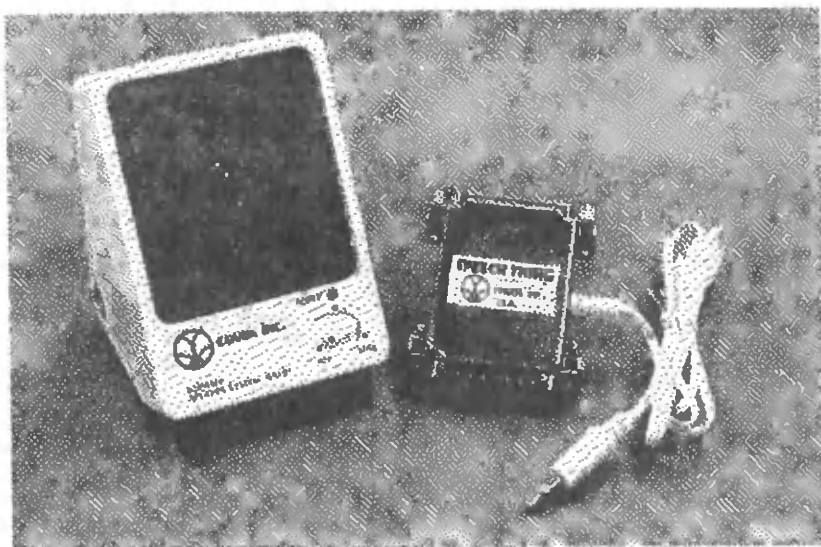
Для слепых и слабовидящих детей, для которых использование зрительного канала получения информации невозможно или затруднено, особое значение имеют программы, обеспечивающие качественный звуковой вывод. Генерация высококачественного звукового сопровождения производится с использованием звуковой карты, приобретение хотя бы одного экземпляра которой является обязательным. Она же позволит в полной мере пользоваться программами мультимедиа.

#### 4. Дополнительное оборудование специального назначения

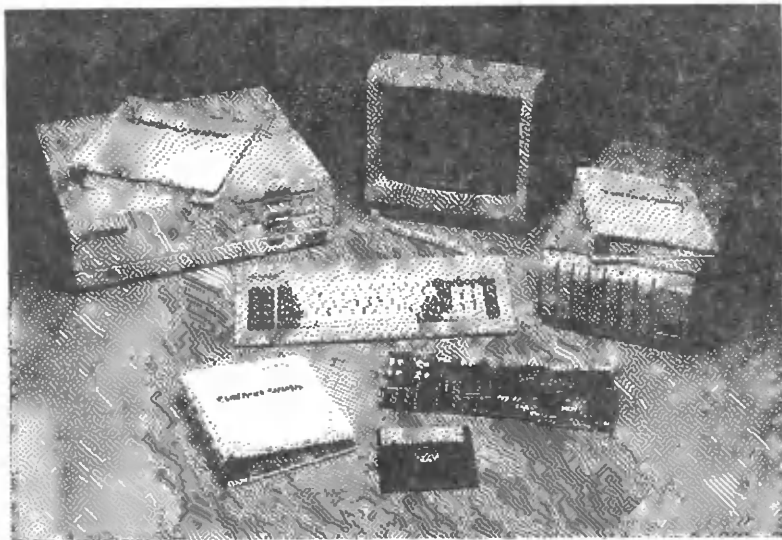
В отличие от рассмотренного выше оборудования, выбор, приобретение, установка и обслуживание которого не вызывает больших вопросов ввиду его широкого распространения, специальные устройства, о которых далее пойдет речь, являются уникальными, предназначенными для использования слепыми и слабовидящими пользователями. Отбор, закупка и установка такого оборудования затруднена отсутствием доступных аналогов, квалифицированных специалистов и методик. Отбор оборудования производился в основном по печатным рекламным и техническим материалам.

##### 4.1. Голосовой синтезатор

Голосовой синтезатор предназначен для совместной работы со специальным программным обеспечением, которое позволяет пользователю воспринимать информацию, отображаемую на экране, на слух. Для слепого пользователя такое оборудование дает возможность без помощи монитора работать с операционной системой и с такими программами, как текстовые редакторы, базы данных, электронные таблицы и т. д. Голосовой синтезатор произносит вслух избранную пользователем на экране информа-



Голосовой синтезатор



#### Голосовой синтезатор

К синтезатору предъявляются следующие требования:

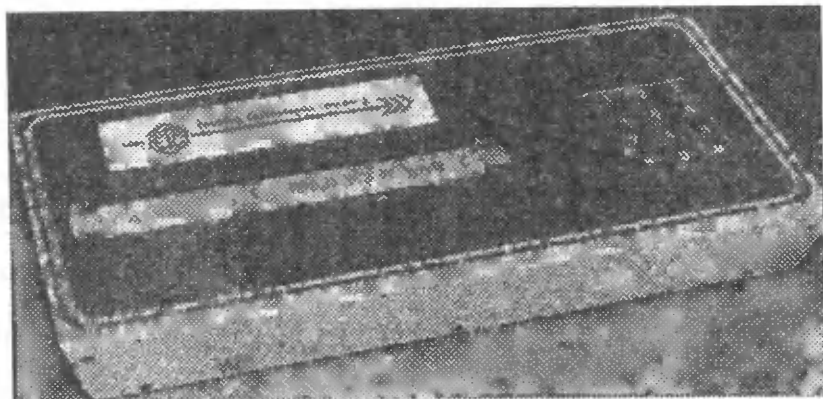
- возможность работы с избранным программным обеспечением;
- качество синтеза голоса;
- возможность чтения текстов на русском языке;
- возможность чтения текстов с фрагментами на русском и английском языках.

Для работы с IBM-совместимой техникой в рамках DOS различными фирмами выпускается большое число различных синтезаторов. Среди них отметим:

- DECTalk PC (фирма DEC);
- Audapter (Personal Data Sys. Inc.);

- Accent — SA (AICOM Corp);
- Keynote Gold SA (Pulsedata Int'l Ltd);
- Echo PC (Echo speech Corp).

Анализ технических характеристик показывает, что наилучшие результаты обеспечиваются первым из перечисленных синтезаторов. Последний же является наиболее дешевым. Детальный анализ технических характеристик, равно как и возможности работы с русскими текстами, может быть проведен лишь после получения подробных технических материалов от фирм-производителей, которые в настоящий момент поступают. Каждое рабочее место ученика и рабочее место учителя необходимо оснастить синтезатором голоса.



Брайлевский терминал



## 4.2. Брайлевский терминал

Брайлевский терминал представляет собой строку из 20—80 брайлевских ячеек, которые дублируют одну из строк дисплея (или часть строки) в виде брайлевского кода. Брайлевские терминалы — очень дорогие устройства. Их использование в известном смысле является альтернативой работе с синтезаторами речи. Если системы речевого вывода экранной информации полезны при проговаривании связного текста, то брайлевские терминалы лучше использовать в тех случаях, когда важна не столько быстрота вывода, сколько его точность. Это происходит, например, при написании программных и других специальных текстов, не являющихся текстами на естественном языке. Пользователь имеет возможность «рассмотреть» каждый выводимый символ, вывести строки экрана в произвольном порядке и в произвольном темпе. Брайлевский терминал также незаменим в тех случаях, когда пользователь работает с форматированным текстом, возникающим на экране. Лишь такой терминал дает незрячему пользователю полную информацию о взаимном расположении различных фрагментов текста.

Среди имеющихся на рынке брайлевских терминалов отметим следующие:

- Alva Braille Terminal (Humanware) 45 cells;
- Alva Braille Terminal (Humanware) 85 cells;
- Braillex IB40 (Adaptec);
- Braillex IB80 (Adaptec);
- INKA (Baum USA);
- KTS Braillotherm (American Thermoform Corp.);

- PowerBraille (TeleSensory).

Сравнительный анализ брайлевских терминалов показывает, что предпочтительным является использование второго из перечисленных устройств. Компьютерный мини-класс предусматривается оснастить одним брайлевским терминалом. При работе учащихся основная ориентация будет делаться на использование синтезаторов речи. Оснащение всех рабочих мест брайлевскими терминалами было бы желательным, однако вряд ли это возможно ввиду их дороговизны. Включение одного терминала позволит учащимся познакомиться и получить навыки работы с ним и тем самым быть готовым к последующему использованию в трудовой деятельности.

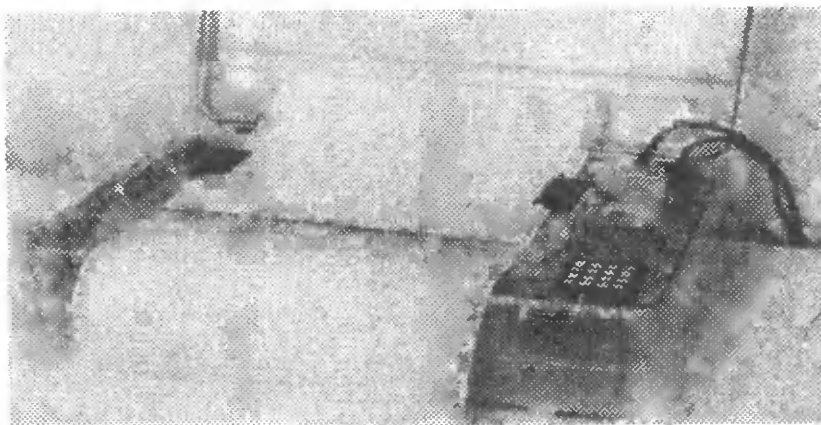
## 4.3. Брайлевский принтер

Брайлевские принтеры позволяют распечатывать информацию в виде брайлевского кода на обычной или специальной бумаге. На рынке предлагается целый ряд брайлевских принтеров, различающихся по следующим параметрам:

- скорость печати;
- размер бумаги,
- односторонняя или двусторонняя печать;
- наличие сервиса для управления незрячим оператором;
- дополнительная возможность обычной черно-белой печати;
- возможность русификации.

Рассматривались следующие принтеры:

- Beta X3 (Thiel GmbH & Comp);
- BAX 10 (Thiel GmbH & Comp);
- PortaThiel (Thiel GmbH & Comp);



Брайлевский принтер

- Everest D (Index);
- Everest S (Index);
- Braillo Commet, 200, 400s (Braillo);

Принтер, установленный в школе-интернате для слепых и слабовидящих детей, безусловно, будет применяться для размножения брайлевских печатных материалов, что накладывает высокие требования на его производительность. Анализ возможностей принтеров и их стоимости показывает, что наилучшим вариантом при разумной цене является принтер Everest D. Он обеспечивает двустороннюю печать на бумаге практически любого качества со скоростью 100 символов в секунду.

## 5. Общее программное обеспечение

В настоящем документе не имеет смысла подробно останавливаться на программном обеспечении общего назначения. Мы лишь перечислим некоторые программные продукты, а точнее категории программных продуктов, наличие которых обязательно:

- операционная система;
- сетевое программное обеспечение Netware 3.12 не менее чем на 10 пользователей, русифицированное;
- текстовый редактор;
- оболочка над операционной системой (например, Norton Comander);
- телекоммуникационное оборудование (Telix, UUCP);
- утилиты различного назначения.

Прочее общее программное обеспечение должно приобретаться и устанавливаться с учетом той конкретной программы, которой конкретный учитель будет придерживаться. Такое программное обеспечение составят некоторые системы программирования, базы данных, электронные таблицы и т. д. В настоящей статье мы не касаемся программного обеспечения, предназначенного для преподавания различных дисциплин (физики, географии, химии, биологии и т. п.). Выбор таких программ и в обычной общеобразовательной школе связан с известными трудностями. При наложении этих трудностей на специфику работы со слабовидящими и слепыми детьми мы попадаем в область педагогики, которая в настоящий момент исследована слабо. Использование накопленных программных материалов должно прово-

диться в экспериментальном порядке и опираться на знания и компьютерную подготовку конкретного преподавателя-предметника.

## 6. Специальное программное обеспечение

Специальное программное обеспечение, которое следует установить в мини-классе, можно разделить на две категории. Программное обеспечение первой категории следует закупить и установить одновременно с установкой оборудования. Программное обеспечение второй категории станет необходимым несколько позже, по мере освоения оборудования.

К первой категории отнесем следующие программы:

- Screen reader. Программа, работающая вместе с синтезатором речи и осуществляющая речевой вывод информации. Таких программ несколько, но следует порекомендовать Screen reader фирмы IBM, опыт работы с которым в России уже накоплен. Необходимо учесть, что для нормальной работы программы необходимо оборудовать компьютеры дополнительной клавиатурой с 18 клавишами
- Текстовый редактор, предназначенный для работы со Screen reader.



Программа увеличения фрагмента экрана



Рабочее место учителя

- Программа, позволяющая увеличивать фрагмент экрана до необходимого размера (Screen Enlargement Program), предназначенная для слабовидящих пользователей. Известно несколько таких программ, но мы рекомендуем MAGic Deluxe фирмы Microsystems Software Inc. Она удобна, универсальна и относительно дешева.
- Программа перевода брайлевского кода (Braille Translation Program).

Мы рекомендуем программу Hot Dots фирмы Raised Dot Computing.

Ко второй категории мы отнесем обучающие программы для слепых пользователей, специальное телекоммуникационное обеспечение, развивающие игры, системы мультимедиа, использование которых возможно рассматриваемым контингентом учащихся. Выбор такого программного обеспечения должен производиться после приобретения учителем и учениками первичного опыта работы с компьютерным классом.

## П. Гангнус

### КРАСОТА ТОЛЩИНОЙ В КОЖУ

Если сегодня вы видите хороший ноутбук, скорее всего у него был активный жидкокристаллический дисплей. Очень вероятно, что в ближайшую пару лет ему появится альтернатива — тоненькие катодные трубки.

Звучит невероятно? Тем не менее на лабораторном уровне технология существует уже лет двадцать пять. Технология называется «Дисплей с ползвым излучателем» или ДПИ (FED). Идея очень проста: вместо того чтобы иметь одну электронную пушку в большой вакуумной трубке, прицельно стреляющую электронами по поверхности, покрытой пятнышками цветных флюорес-

центов, ДПИ помещает за каждым флюоресцирующим пятном несколько тысяч крохотных излучателей электронов.

Таким образом, качество изготовления становится значительно менее критичным, чем в LCD-экранах. В них пара неисправных транзисторов может испортить весь экран. В ДПИ может выйти из строя более половины эмиттеров, а вы будете видеть все то же изображение.

ДПИ имеет и другие преимущества — такие, как более быстрое время реагирования, что полезно для анимации и видео, более низкое энергопотребление и более широкий угол нормальной видимости.

Несколько маленьких компаний пытаются внедрить ДПИ на рынок. У каждой — свой подход. Pixel в этом году собирается представить 10-дюймовый прототип.

Один из пионеров — SI Diamond Technology (Хьюстон, Техас) — применяет для создания эмиттеров тонкую алмазную пленку. Это существенно упрощает производство литографических работ и позволяет использовать низкое напряжение. Компания уже объявила об успехе в создании низковольтных флюоресцентов.

Пожоже, мы увидим прототипы в этом году и компьютеры на продажу в 1997. Да будут ваши экраны тоньше и ярче!



Министерство образования Российской Федерации  
рекомендует учебно-методический комплект КуМир  
в качестве основного учебного материала

**Пока еще есть в продаже**

ЯмахаМир - 480000 рублей, КорветМир - 480000 рублей,  
КуМир УКНЦ - 540000 рублей

### Заявка

Я хочу приобрести программные средства  
Объединения ИнфоМир: \_\_\_\_\_

Счет выслать по адресу: \_\_\_\_\_

*Заявки направлять по адресу: 103051, Москва,  
Садовая-Сухаревская, д.16, к.9 "Информатика и образование"*  
Для Объединения ИнфоМир

Copyright Объединение ИнфоМир © МСМХСУ  
ИнфоМир, логотип Объединения ИнфоМир, КуМир, КуМир-Гипертекст,  
Мультимедиа-КуМир, ПлацМир, НовоМир, МикроМир, КорветМир,  
ЯмахаМир - торговые марки Объединения ИнфоМир

## Объединение ИнфоМир

ведущая и старейшая российская компания в области новых  
информационных технологий в образовании и  
программно-методического обеспечения по информатике  
предлагает Вашему вниманию

### КуМир-Гипертекст (DOS, Mac)

*базовый комплект, расширенный комплект,  
Мультимедиа-КуМир, ПлацМир*

### НовоМир (DOS, Mac)

*программно-методическая гипертекстовая система для  
преподавания в 8, 9 и 10 классах, включающая разделы:  
базы данных, электронные таблицы, сортировка и поиск, кодирование  
информации, передача информации, информационные модели*

### МикроМир (DOS, WIN, Mac)

*текстовый процессор и электронные таблицы  
учебного и производственного назначения*

Новые методики и программные продукты

Объединения ИнфоМир,  
опыт работы в системе КуМир-Гипертекст,  
ответы на Ваши вопросы и многое другое в нашем  
бесплатном электронном списке рассылки.

Отправляйте Ваши послания по адресу  
**GMK@nw.math.msu.su**

Наиболее интересные сообщения  
будут рассылаться подписчикам списка

Shareware и Demo-версии программных продуктов

Объединения ИнфоМир находятся на сервере  
механико-математического факультета

Московского Государственного Университета

**URL: ftp://mech.math.msu.su/pub/InfoMir**

Пароль для входа: **anonymous**

Объединение



ИнфоМир

# МЕТОДИКА

**Л. А. Залогова,**

*кандидат физико-математических наук, Пермский государственный университет,  
Лаборатория информатизации образования Пермской области*

**И. Г. Семакин,**

*кандидат физико-математических наук, Пермский государственный университет,  
Лаборатория информатизации образования Пермской области*

## УРОКИ ПО WINDOWS\*

### Урок 2. Работаем с программами

Цель данного урока: научиться управлять работой прикладных программ, пользоваться услугами Диспетчера программ.

#### Основные понятия

**Приложения Windows.** Так называют программы, созданные для работы в системе Windows.

**Диспетчер программ (ДП).** Программа, занимающая центральное место в работе с системой. ДП управляет работой всех прочих программ.

При запуске Windows ДП загружается автоматически и на экране появляется его окно. С окном ДП можно делать то же, что и с любыми другими окнами. Его можно увеличивать, перемещать, уменьшать, сворачивать в пиктограмму и пр. От других программ ДП отличаются две главные особенности:

- все прочие программы запускаются с помощью ДП;
- пока Windows работает, ДП активен; выход из ДП заканчивает сеанс работы с Windows.

**Группы программ.** Все приложения Windows объединяются в группы. Каждой группе присваивается собственное имя. Группа может включать не более 20 программ. Обычно программы объединяют по принципу общности их назначения. Например, группа **Игры** включает в себя только игровые программы.

Каждая группа имеет собственное окно, так называемое **групповое окно**, в котором располагаются символы программ, входящих в данную группу:



Групповое окно можно уменьшить до размера символа. Символ группового окна имеет форму прямоугольника, в котором изображены шесть крошечных символов программ. Под прямоугольником записывается имя группы:



Игры

\* Продолжение. Начало см.: Информатика и образование. 1995. № 4.

После запуска Windows по умолчанию создаются несколько групп (Главная, Реквизиты, Игры и др.). Символы их групповых окон располагаются в окне ДП.

Манипулирование групповыми окнами (раскрытие, сворачивание в пиктограмму, перемещение, изменение размеров) производится точно так же, как и с окнами прикладных программ.

**Стандартные приложения Windows.** В стандартной поставке Windows находится несколько приложений. Это небольшие по размеру и возможностям прикладные программы, необходимые для повседневной работы. В их состав входят:

- графический редактор Paintbrush;
- текстовый редактор Write;
- картотека;
- календарь;
- часы;
- игры и др.

**Paintbrush** — это программа для создания рисунков [1].

**Write** — текстовый редактор [2], уступающий по возможностям «большим» профессиональным программам обработки текстов. Однако для несложных работ Write оказывается полезным и достаточно эффективным.

Картотека состоит из карточек, на которых можно хранить текст и картинки. Каждая карточка состоит из области ввода и специальной строки для ключевого слова. Например, при создании картотеки адресов работников фирмы в качестве ключевого слова удобно выбрать фамилию. Поиск карточки легко осуществить, если известно ключевое слово. Однако карточку можно отыскать не только по ключевому слову. Программа **Картотека** может найти любую информацию на любой карточке.

Программа **Календарь** представляет собой календарь — памятку для ежедневной работы. Можно составить план дня, а также спланировать работу на месяц. В **Календаре** предусмотрена специальная возможность, называемая функцией сигнализации (или просто «будильник»), которая напомнит вам о наступлении важных моментов времени. Календарем можно эффективно пользоваться лишь тогда, когда компьютер включен «с утра до вечера».

В наше время трудно представить рабочий стол без **Калькулятора**. В состав системы Windows входит два калькулятора — стандартный и научный. Стандартная версия выполняет обычные функции простого калькулятора (сложение, вычитание, умножение, деление, извлечение квадратного корня и др.). Научный калькулятор предоставляет возможность работать с разнообразными функциями (вычисление среднего, суммы величин, поразрядные логические операции, тригонометрические функции и др.), а также использовать четыре системы счисления.

**Часы** показывают внутреннее время компьютера (системное время). Часы могут быть представлены в одной из двух форм — цифровой или стрелочной. В цифровом представлении высвечиваются часы, минуты, секунды и дата. С помощью специальной команды можно сменить вид циферблата на стрелочный.

**Справочник Windows.** Любое современное фирменное программное обеспечение содержит в себе справочную информацию (**Справочник, Help**). Справочник дает возможность пользователю оперативно получать необходимые сведения по работе с системой. Справочник является своеобразным электронным учебником с гипертекстовым принципом организации материала.

В справочник Windows можно попасть через меню **Справочник** в строке меню **Диспетчера программ**. Научиться работе со справочником несложно. Для этого в нем имеются все необходимые пояснения.

#### **Упражнение 1. Манипулирование групповыми окнами.**

**Задание 1.** Запустить Windows ([3], упражнение 1).

На экране появится окно **Диспетчера программ** с расположенными в нем символами групповых окон.

Используя приемы, освоенные в предыдущем уроке, проделать следующее:

- > перемещать пиктограммы групповых окон по экрану;
- > открыть одно из групповых окон;

- > перемещать, изменять размеры группового окна (убедитесь в том, что групповое окно можно перемещать только в пределах окна Диспетчера программ).

**Упражнение 2. Работа со строкой меню.**

Словом *меню* называют набор пунктов, выбираемых из списка (рис. 1). Пункт меню может представлять собой исполняемую команду или имя другого меню, которое раскрывается на экране при его выборе. Использование меню — один из основных приемов управления работой приложений в Windows.

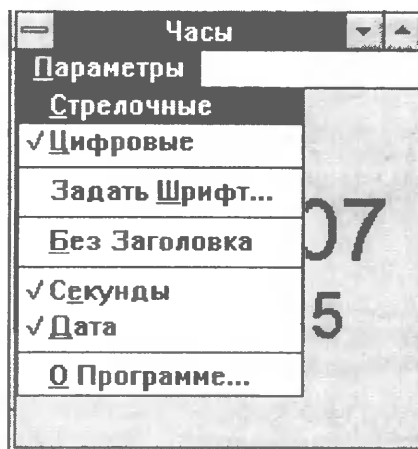


Рис. 1. Пункты меню Параметры программы Часы

**Задание 1.** Запустить прикладную программу Часы:

- > открыть групповое окно Реквизиты;
- > дважды щелкнуть левой клавишей мыши на символе программы Часы. Вы увидите, что строка меню содержит одно меню — Параметры.

**Задание 2.** Изменить внешний вид (форму) часов:

*1-й способ.* Выбор пункта меню с помощью нажатой кнопки мыши (нажми и держи):

- > в строке меню подвести курсор мыши к меню Параметры;
- > нажать левую клавишу мыши и держать ее нажатой (меню раскроется);
- > перемещать мышь с нажатой клавишей до тех пор, пока не будет выделена команда Цифровые или Стрелочные;
- > отпустить клавишу мыши (команда будет выполнена).

*2-й способ.* Выбор пункта меню посредством щелчка (нажми и отпусти):

- > в строке меню подвести курсор мыши к меню Параметры;
- > щелкнуть левой клавишей мыши (меню раскроется);
- > щелкнуть левой клавишей мыши на нужном пункте меню (команда будет выполнена).

**Задание 3.** Для часов в цифровой форме включить/выключить индикацию секунд и даты:

- > выполнить команду Секунды меню Параметры (одним из двух способов, описанных в задании 2);
- > выполнить команду Дата меню Параметры.

*Замечание.* Если включен вывод секунд (или даты), то соответствующий пункт меню помечен галочкой.

**Задание 4.** Яркие и неяркие пункты меню:

- > для стрелочных часов раскройте меню Параметры. Вы увидите, что команда Задать шрифт написана неярко (если создается такая ситуация, когда некоторый

пункт меню не может быть активизирован, Windows пишет название этого пункта неярко);

- > измените форму часов на цифровую и снова взгляните на пункт меню **Задать шрифт** (теперь название пункта написано ярко и он может быть активизирован).
- З а д а н и е 5.** Управление порядком расположения групповых окон.  
 Такое управление производится с помощью меню **Окно Диспетчера программ**.
- > открыть групповые окна **Реквизиты, Главная, Игры**;
  - > в строке меню **ДП** подвести курсор мыши к меню **Окно**;
  - > щелкнуть левой клавишей мыши (меню раскроется);
  - > щелкнуть левой клавишей мыши на пункте **Каскад** (открытые групповые окна наложатся друг на друга таким образом, что прямоугольники всех заголовков одновременно будут видны на экране);
  - > выполнить команду **Мозаика** меню **Окно** (сравните расположение групповых окон в результате исполнения команд **Каскад** и **Мозаика**).

### Упражнение 3. Работа с диалоговыми окнами.

Если команда, отданная через меню, не может быть выполнена без дополнительной информации, то такая информация запрашивается через диалоговое окно.

Диалоговые окна содержат следующие поля управления:

- поле списка;
- командные кнопки;
- поле ввода;
- квадратные поля выбора варианта;
- круглые поля выбора варианта.

**З а д а н и е 1.** Изменить шрифт для цифровых часов (использование поля списка и командных кнопок):

- > открыть окно программы **Часы** (если вы уже закрыли его);
- > установить цифровую форму часов;
- > выполнить команду **Задать шрифт** меню **Параметры**.

Откроется диалоговое окно:

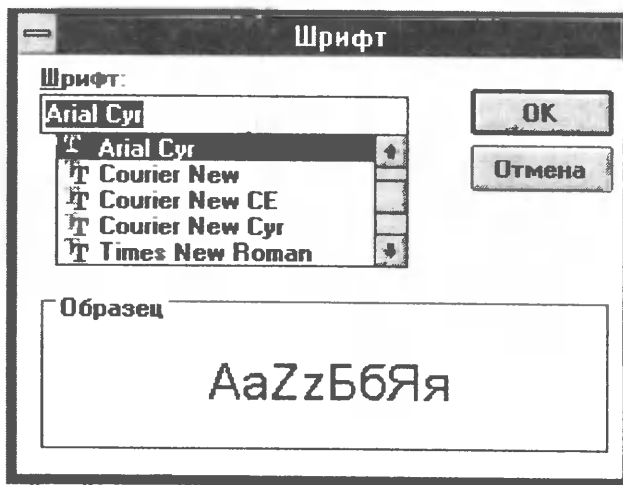


Рис. 2. Диалоговое окно для установки вида шрифта программы **Часы**

В диалоговом окне имеется поле списка, в котором перечислены все имеющиеся в распоряжении шрифты для цифровой формы часов. Выбранный в текущий момент шрифт занесен в прямоугольник над списком. Так как список шрифтов достаточно длинный, в правой части списка предусмотрена линейка прокрутки;



- > щелкнуть левой клавишей мыши на выбранном варианте шрифта.
- В правой части окна на рис. 2 расположены командные кнопки **ОК** (подтвердить) и **Отмена**. Нажав на кнопку **ОК**, вы выйдете из диалогового окна с подтверждением выбора шрифта. С помощью командной кнопки **Отмена** также осуществляется выход из диалогового окна, однако сделанный выбор при этом отменяется;
- > щелкнуть левой клавишей мыши на командной кнопке **ОК** для подтверждения выбора шрифта;
- ИЛИ
- > щелкнуть левой клавишей мыши на командной кнопке **Отмена**, если вы передумали устанавливать новый шрифт;
  - > закрыть окно программы **Часы** (дважды щелкнуть левой клавишей мыши на кнопке **Системного меню**).

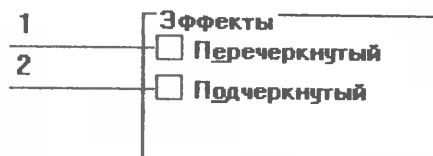
**З а д а н и е 2.** Установить новый шрифт в графическом редакторе Р (использование квадратных полей выбора варианта, поля списка и командных кнопок):

- > запустить **Paintbrush**;
- > щелкнуть левой клавишей мыши на текстовом символе в таблице инструментов (в левой части окна графического редактора);
- > щелкнуть левой клавишей мыши в той части экрана, где должен быть расположен текст;
- > набрать любой текст;
- > в строке меню выбрать меню **Текст**;
- > выбрать пункт меню **Шрифты**.

В открывшемся диалоговом окне:

- > выбрать с помощью полей списков шрифт, вид и размер шрифта (щелкнуть левой клавишей мыши на выбранном варианте каждого поля списка).

Кроме того, можно реализовать два специальных эффекта: *перечеркивание* и *подчеркивание*. Для этого используются *квадратные поля выбора*. Например,



Можно включить (позначить) любое количество квадратных полей выбора. При включении поле выбора содержит символ **X**, в противном случае — пусто. Квадратные поля выбора не исключают друг друга и, следовательно, могут включаться одновременно;

- > установить символ **X** в квадрат 1, щелкнув левой клавишей мыши в этом квадрате;
- > установить символ **X** в квадрат 2;
- > щелкнуть левой клавишей мыши на командной кнопке **ОК**;
- > набрать любой текст (текст будет подчеркнут и перечеркнут).

Теперь наберите текст, который будет только подчеркнут или только перечеркнут.

**З а д а н и е 3.** Установить размер рисунка и палитру (использование **Круглых полей выбора варианта** и **Командных кнопок**).

Круглые поля выбора выполняют роль переключателя между взаимоисключающими вариантами. Например, в **Paintbrush** с помощью команды **Атрибуты образа** меню **Параметры** можно определить особые характеристики рисунка (размер и палитру);

- > в окне **Paintbrush** активизировать команду **Атрибуты образа** меню **Параметры**.

Откроется диалоговое окно, в котором расположены две группы из круглых полей выбора (рис. 3).

Пользователю предоставляется возможность использовать черно-белую и цветную палитру, а также регулировать размеры рисунка.

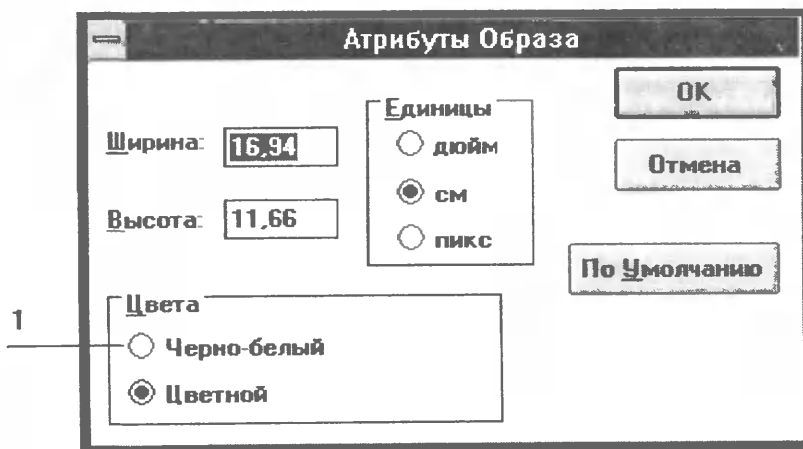


Рис. 3. Диалоговое окно для установки атрибутов изображения

- > щелкнуть левой клавишей мыши по круглому полю 1;
- > щелкнуть левой клавишей мыши по командной кнопке **ОК**.  
Перед вами — черно-белая палитра.

**Задание 4.** Изменить размер прямоугольника, в котором будет создаваться рисунок (использование круглых полей выбора варианта и командных кнопок):

- > в окне **Paintbrush** активизировать команду **Атрибуты образа** меню **Параметры**.  
Откроется диалоговое окно. С помощью круглых полей выбора, заключенных в рамку **Единицы**, можно изменить единицу измерения для задания размера прямоугольника, в котором будет создаваться рисунок;
- > выбрать единицу измерения, щелкнув левой клавишей мыши по какому-нибудь круглому полю;
- > установить высоту и ширину рисунка, используя поля ввода **Высота** и **Ширина**;
- > щелкнуть левой клавишей мыши на кнопке **ОК**.

Создаваемый рисунок появится в прямоугольнике установленного размера.

**Задание 5.** Сохранить изображение в файле на диске (использование поля ввода):

- > в окне графического редактора **Paintbrush** нарисовать любое изображение;
- > выполнить команду **Сохранить как..** (сохранить в файле с именем). Откроется диалоговое окно, содержащее поле ввода:

**Имя Файла:**

- > указать имя файла в поле ввода **Имя файла**;
- > щелкнуть левой клавишей мыши по командной кнопке **ОК** (изображение запомнится в файле с указанным именем).

**Упражнение 4. Параллельная работа приложений.**

Наиболее яркая и отличительная особенность **Windows** — возможность одновременного запуска и параллельной работы нескольких приложений.

Параллельная работа с несколькими программами во многих случаях обеспечивает пользователю удобную рабочую обстановку. Например, при составлении отчета необходим редактор текста (для ввода текста отчета), калькулятор (для вычислений), а также база данных (для выборки сведений о клиентах). Причем все эти программы необходимы в активном состоянии, чтобы время от времени быстро переключаться между ними.

Операционная система MS DOS является однопользовательской и однозадачной, т. е. она рассчитана на выполнение в каждый конкретный момент только *одной* программы *одного* пользователя. Windows предоставляет возможность параллельного выполнения *нескольких* программ *одного* пользователя. Для работы *многих* пользователей со *многими* программами необходимы сетевые средства и/или другая операционная система (например, UNIX).

Параллельное выполнение нескольких программ часто называют *многозадачностью*, рассматривая каждую выполняемую программу как отдельную задачу.

Любой запуск приложения из ДП является свидетельством многозадачности, так как ДП и приложение — это разные программы, параллельно выполняемые в Windows.

Каждая запущенная программа высвечивает на экране свое окно (рис. 4).

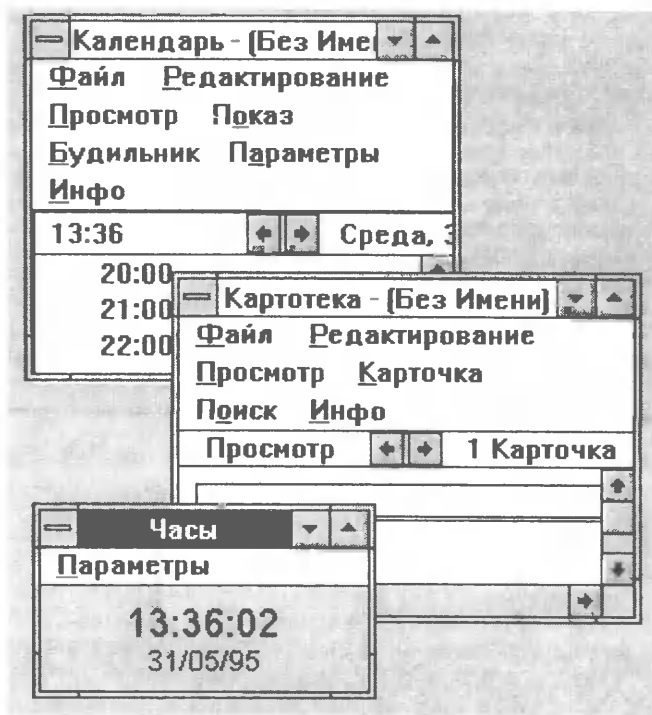


Рис. 4. Окна параллельно запущенных программ

Среди окон параллельно работающих программ существует только одно активное окно. Это окно располагается над всеми остальными окнами, и только с ним взаимодействуют мышь и клавиатура. Самым простым способом активизации окна является щелчок левой клавишей мыши в области этого окна ([3], упражнение 4). Активизируемое окно перерисовывается над всеми окнами, а его прямоугольник заголовка и рамка приобретают окраску, характерную для активного окна. В Списке задач имя активной программы также выделяется цветом.

**З а д а н и е 1.** Преобразовать окна работающих программ в пиктограммы.

В случае дефицита экранного пространства окно работающей программы может быть преобразовано в пиктограмму, при этом приложение не выгружается из памяти и не прекращает свою работу.

- > открыть групповое окно Реквизиты (если вы его закрыли);
- > запустить программы Write, Paintbrush, Часы, Калькулятор;
- > преобразовать окно Write в пиктограмму, щелкнув левой клавишей мыши по Уменьшителю;

- > преобразовать окно **Paintbrush** в пиктограмму;
- > преобразовать окно **Часов** в пиктограмму;
- > преобразовать окно **Калькулятора** в пиктограмму;
- > преобразовать окно **ДП** в пиктограмму. В результате этих действий в нижней части экрана расположится пять пиктограмм.

**З а д а н и е 2.** Выполнить переключение между приложениями.

Переход от одного приложения к другому называется *переключением*. Когда запущено несколько программ, появляется необходимость быстрого переключения между ними. Это удобно, например, при обмене данными. Рассмотрим в качестве примера переключение с текстового редактора **Write** на графический редактор **Paintbrush**.

*1-й способ:*

- > дважды щелкнуть левой клавишей мыши на пиктограмме **Write** (так как текстовый редактор **Write** запущен, то для открытия его окна достаточно выполнить двойной щелчок на его пиктограмме);
- > дважды щелкнуть левой клавишей мыши на пиктограмме **Paintbrush**. Откроется окно графического редактора.

*2-й способ* (удобен в тех случаях, когда одно или несколько приложений работают в окнах с полноэкранным представлением):

- > активизировать окно **Write**;
- > увеличить размеры **Write** до размеров экрана (теперь пиктограммы работающих программ, а также окно **Paintbrush** не видны);
- > нажать на клавишу **Alt** и держать ее нажатой;
- > нажать на клавишу **Tab** (в окне **Write** появится маленькое информационное окно с изображением пиктограммы и именем приложения);
- > нажать на клавишу **Tab** еще раз, продолжая удерживать **Alt** (содержимое информационного окна изменится: здесь появится ссылка на следующее работающее приложение);
- > нажимать **Tab** до тех пор, пока в информационном окне не появится ссылка на интересующее вас приложение;
- > отпустить клавишу **Alt**.

*3-й способ* (удобен при большом числе параллельно работающих приложений, так как позволяет достичь цели быстрее, чем переключение по кругу с помощью **Alt+Tab**):

- > активизировать **Список задач**;
- > дважды щелкнуть левой клавишей мыши на имени той задачи, на которую требуется переключиться

**ИЛИ**

один раз — на имени и один раз на кнопке **Переключиться в**.

**З а д а н и е 3.** Многократный запуск приложения.

Приложения, предназначенные для обработки некоторых документов (текстов, иллюстраций и др.), можно запускать многократно для параллельной работы. В этом случае каждый экземпляр приложения работает со своим документом.

- > Запустить **Write** (это будет второй запуск);
- > переключиться на первый экземпляр **Write**.

**З а д а н и е 4.** Упорядочить окна на экране.

Рекомендуем повторить упражнение 5 из [3].

**Упражнение 5.** Организация обмена данными между приложениями.

При работе с несколькими программами можно передавать информацию из одной программы в другую. С помощью такого перемещения информации можно объединить работу, сделанную в нескольких программах, в один документ. Например, нужно ввести в текстовый документ, созданный **Write**, графическое изображение, полученное с помощью **Paintbrush**. Для этого надо *выделить* нужную информацию в одной из программ, *скопировать* ее в промежуточный карман и затем *вставить* ее в другую программу.

*Промежуточный карман* — область памяти, которую **Windows** предоставляет в распоряжение всех программ. Данные, сохраненные в промежуточном кармане одной программой, могут быть считаны другой программой. Промежуточный карман

можно использовать только для одного сохранения. Каждое последующее занесение информации в промежуточный карман уничтожает его старое содержимое без предупреждения.

**Задание 1.** Скопировать изображение, созданное в Paintbrush, в промежуточный карман:

- > запустить программу Paintbrush;
- > нарисовать любую картинку;
- > вырезать картинку (или ее фрагмент) с помощью ножниц;
- > выполнить команду **Вырезать** меню **Редактирование** (изображение скопируется в промежуточный карман).

**Задание 2.** Включить графическое изображение из промежуточного кармана в документ, созданный Write:

- > запустить программу Write;
- > набрать произвольный текст;
- > отметить левой клавишей мыши место вставки картинки;
- > выполнить команду **Вставить** меню **Редактирование**.

Таким образом, получен документ, содержащий текст и иллюстрацию.

**Задание 3.** Внести изменения в иллюстрацию, вставленную в текстовый документ:

- > дважды щелкнуть левой клавишей мыши на рисунке, созданном в предыдущем задании в окне программы Write.

Открывается окно программы Paintbrush. Открывшееся окно располагается на переднем плане, и в него загружается выбранное щелчком изображение. Таким образом, в окне графического редактора теперь находится фрагмент Write-документа.

- > внести изменения в рисунок;
- > выполнить команду **Обновить** меню **Файл** (изменения будут перенесены в рисунок программы Write);
- > завершить исполнение Paintbrush.

Перед вами — Write-документ с измененной иллюстрацией.

#### **Упражнение 6. Работа с окном буфера обмена.**

Обмен данными через промежуточный карман протекает скрыто от пользователя, поэтому интересно знать, какая именно информация в нем хранится. В Windows имеется специальная программа **Окно буфера обмена**, которая позволяет просмотреть промежуточный карман, сохранить его в файле (или загрузить из файла), а также удалить его содержимое.

**Задание 1.** Просмотреть содержимое промежуточного кармана:

- > открыть групповое окно **Главная**;
- > дважды щелкнуть на пиктограмме **Окно буфера обмена** (содержимое промежуточного кармана появится в окне программы **Окно буфера обмена**).

**Задание 2.** Удалить содержимое промежуточного кармана:

- > в строке меню окна буфера обмена открыть меню **Редактирование**;
- > выполнить команду **Удалить**.

Кроме того, с помощью команд меню окна буфера обмена можно сохранить содержимое кармана в файле и восстановить из файла.

#### **Упражнение 7. Управление группами программ.**

Управление группами программ осуществляет **Диспетчер программ**. С помощью ДП можно создать новую группу, добавить в группу новую программу, удалить программу из группы; уничтожить группу.

**Задание 1.** Создать новую группу:

- > активизировать ДП;
- > выполнить команду **Создать** меню **Файл Диспетчера программ**.

Откроется диалоговое окно **Новый программный объект**;

- > включить в диалоговом окне круглое поле выбора **Группа программ** (если оно не включено).

Откроется диалоговое окно **Свойства группы программ**;

- > ввести имя новой группы в поле **Описание**, например «example».
- Теперь можно заполнять группу символами программ.

**Задание 2.** Включить новую программу в группу (программа должна быть разработана специально для работы с Windows):

- > активизировать **Диспетчер программ** (если он не активизирован);
- > открыть групповое окно, в которое будет включена программа (если оно закрыто);
- > выполнить команду **Создать меню Файл Диспетчера программ**.
- Откроется диалоговое окно **Новый программный объект**;
- > включить круглое поле **Программный элемент**.
- Откроется диалоговое окно **Свойства программного элемента**;
- > нажать кнопку **Просмотреть**;
- > выбрать имя файла с помощью полей **Устройство**, **Каталог** и поля списка **Имя файла**.

**Задание 3.** Удалить символ программы из группы:

- > активизировать **Диспетчер программ** (если он не активизирован);
- > открыть групповое окно, из которого необходимо исключить символ программы (если оно закрыто);
- > пометить в групповом окне символ программы;
- > нажать на клавишу Delete.

Появится диалоговое окно, с помощью которого Windows должна получить подтверждение об удалении символа программы из группы. Программа не стирается с диска, а только удаляется из группы.

**Задание 4.** Удалить группу:

- > уменьшить групповое окно до пиктограммы;
- > пометить символ группового окна;
- > нажать на клавишу Delete.

В результате стирания группы все символы программ, входящих в группу, также стираются. Однако программы сохраняются на диске и могут быть включены в другую группу.

## Литература

1. Залогова Л. А., Семакин И. Г. Компьютер рисует//Информатика и образование. 1995. № 1.
2. Русаков С. В., Семакин И. Г. Компьютер пишет//Информатика и образование. 1994. № 6.
3. Залогова Л. А., Семакин И. Г. Уроки по Windows//Информатика и образование. 1995. № 4.
4. Фаненштих К., Хаселер Р. Операционная среда Windows. М.: Изд. ЭКОМ, 1994.

### *Уважаемые читатели!*

напоминаем вам, что с продолжается подписка на первое полугодие 1996 г. Сведения, необходимые для подписки на журнал «Информатика и образование»:

|                                | Индекс издания | Цена издания по каталогу в руб. | Периодичность    |
|--------------------------------|----------------|---------------------------------|------------------|
| для индивидуальных подписчиков | 70423          | 36 000<br>(за три номера)       | 1 раз в 2 месяца |
| для предприятий и организаций  | 73176          | 105 000<br>(за три номера)      |                  |

**ОБРАЩАЕМ ВАШЕ ВНИМАНИЕ, ЧТО НА ЖУРНАЛ  
МОЖНО ПОДПИСАТЬСЯ ПО БЕЗНАЛИЧНОМУ РАСЧЕТУ**

Телефон для справок: (095) 208-30-78

Факс: (095) 208-67-37

E-Mail: info@tit-bit.msk.su

**В. В. Грамолин,**

*директор отдела информатизации Фонда развития парламентаризма в России*

## **ХОЧУ РАБОТАТЬ В WINDOWS!** **(Уроки работы в текстовом редакторе** **Word for Windows 6.0 русская версия)**

Если у вас есть компьютер, на котором установлены Windows, и кроме того текстовый редактор Word for Windows; если рядом с вами есть специалист или «Сосед», готовый иногда помочь вам, но вам неудобно дергать его каждую минуту и хочется шаг за шагом освоить этот инструмент самостоятельно; при наличии всего этого — вперед, мы проведем вас по этому пути!

Настоящая публикация — первая в серии планируемых публикаций. Это не учебник, не руководство, а это попытка провести за руку совершенного новичка: сначала научить его не бояться компьютера, потом помочь овладеть конкретными навыками, может быть самыми простыми, а дальше научить его работать самостоятельно и постепенно углублять свои знания. Дальнейшее обучение, а вернее сказать углубление знаний, можно проводить по книгам, руководствам и справочникам, в том числе и по «Справочнику», расположенному в самой программе текстового редактора Word for Windows, однако при работе со «Справочником» следует понимать, что он написан не столько для первоначального изучения, сколько для того, чтобы вспомнить что-либо подзабытое из ранее изученного.

Начать работу лучше всего с чтения названий уроков. Уроки выстроены в таком порядке, чтобы овладение текстовым редактором происходило постепенно. Прервать или приостановить обучение можно на любом этапе. Даже пройдя только первый урок, уже можно получить определенные знания и опираться на них в практической работе.

Настоящее руководство посвящено рассмотрению конкретных задач, возникающих при работе с Windows и с текстовым редактором Word. В то же время некоторые простейшие навыки работы в Windows можно освоить самостоятельно, открыв программу Учебник по Windows. Эта программа, громко названная учебником, позволит вам потренироваться в работе с мышью, что на первых порах очень важно, и покажет основные приемы в работе с окнами. Если у вас нет навыка работы с мышью, предлагается эту небольшую часть Учебника по Windows изучить (ниже сказано, как это можно сделать) и только потом приступить к первому уроку. Вторую часть Учебника по Windows рекомендуется просмотреть и освоить либо сразу после первого урока, либо непосредственно перед вторым уроком.

Итак, если есть необходимость потренироваться в работе с мышью, откроем Учебник по Windows. Для этого необходимо запустить Windows и открыть окно Диспетчера Программ. Если вам это сделать затруднительно, то попросите помощи у «Соседа», если возможно это сделать самим, тогда ваши действия должны быть следующими:

- установите указатель мыши (на экране он представлен в виде стрелки) на верхнюю строку меню на слово Справка и нажмите левую кнопку мыши;
- в выпавшем меню установите указатель мыши на слова Учебник по Windows и нажмите левую кнопку мыши;
- далее следуйте указаниям на экране, но сначала освоите только работу с мышью, после чего покиньте этот учебник и вернитесь к нему для освоения работы с окнами лишь после первого урока.

Итак, шаг за шагом (step by step) будем овладевать системой Windows и текстовым редактором Word for Windows.

## Урок 1. Как сразу написать письмо, распечатать его на бумаге, а затем сохранить для будущего архива?

Если вам необходимо написать письмо или просто объявление, вы запускаете на компьютере сначала Windows, а потом Word\* и через некоторое время видите на экране множество кнопок, меню, линеек и т. п. (рис. 1.1). Если на вашем экране не хватает чего-либо по сравнению с рисунком, не огорчайтесь, — может быть, это и не нужно, но на всякий случай можно посоветоваться с «Соседом». А лучше давайте сразу «ввяжемся в бой».

Считайте, что перед вами обыкновенная пишущая машинка. На экране чистое пространство, на котором можно набирать текст. На этом пространстве выделяется мигающая вертикальная палочка, называемая курсором. Он указывает место, с которого начинается заполнение экрана буквами.

Наберите текст письма, представленного на рис. 1.1. На этом рисунке показан текст, который нуждается в последующем оформлении. Такое оформление (называемое форматированием) мы научимся делать на третьем уроке. Но уже сейчас вы можете посмотреть, как будет этот текст выглядеть после форматирования (рис. 3.1). А пока будем просто набирать этот текст, а потом распечатаем.

В отличие от пишущей машинки при наборе текста на компьютере перевод на другую строку происходит автоматически. Текст, набираемый на клавиатуре, сразу отображается на экране, а место печати очередного символа обозначено курсором. Если в процессе набора возникает ошибка, то ее можно исправить:

- для исправления буквы, стоящей перед курсором, нужно нажать клавишу BackSpace (обычно на этой клавише нарисована стрелка ←);
- для исправления буквы, стоящей после курсора, нужно нажать клавишу с надписью Delete или Del.

При наборе вам придется переключаться на латинскую клавиатуру и обратно: это делается с помощью произвольно установленной комбинации клавиш, например Ctrl и Shift, возможны и другие сочетания: Shift+Shift (т. е. одновременное нажатие двух клавиш Shift (левой и правой) либо только одной клавиши, например левой Ctrl). Каждое нажатие на такую клавишу переключает клавиатуру с латиницы на кириллицу или наоборот.

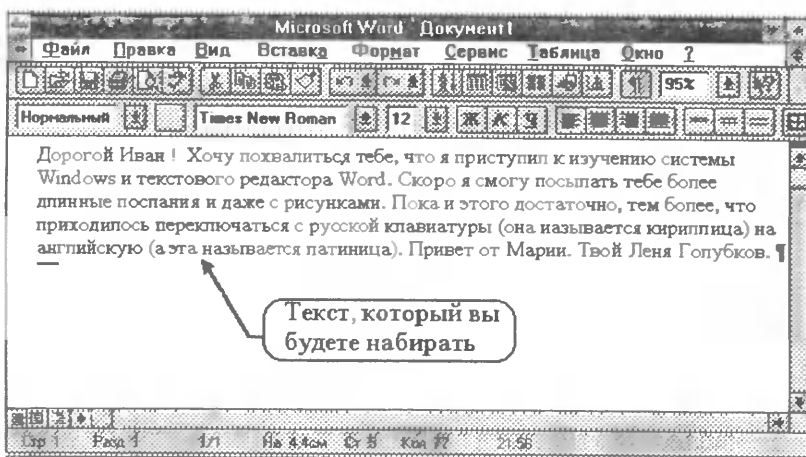


Рис. 1.1

\* Запуск Windows обычно выполняется путем набора на клавиатуре win и затем нажатием клавиши Enter. Запуск Word (после того как Windows будет установлен) производится двойным нажатием на иконке Word. Если на этом этапе что-либо не получается — попросите помочь «Соседа».



Теперь вперед — печатайте либо то письмо, что приведено на рис. 1.1, либо придумайте свое.

Нужно заметить, что вид текста на вашем экране может несколько отличаться от приведенного на рисунке, поскольку текстовый редактор мог быть заранее настроен, например, на ввод текста с красной строки либо на иной размер шрифта, чем показан на рисунке. Все эти вопросы оставим до третьего урока, а пока наша задача — набрать и напечатать текст.

Итак, текст напечатан. Упорядочим его и приведем в соответствие с рис. 1.2.

Для выполнения этого задания необходимо сделать следующее:

- поставьте указатель мыши после первого предложения и нажмите левую кнопку. На этом месте будет установлен курсор экрана;
- нажмите клавишу Enter. В результате этого нажатия следующее предложение будет начинаться с новой строки.

В конце предыдущей строки установился знак конца абзаца. Знак этот будет виден на экране, если в стандартной панели (рис. 1.2) найти и «нажать» на кнопку Спец. символы (на этой кнопке нарисован знак конца абзаца ¶). Еще одно «нажатие» на эту кнопку — и знаки с экрана исчезнут. «Нажатием» на кнопки на экране мы будем называть следующую процедуру: указатель мыши подводится к кнопке на экране, которую нужно нажать, нажатие происходит с помощью левой кнопки мыши.

Повторим операцию установки абзаца для второго предложения, т. е. установим курсор перед словом «скоро» и нажмем клавишу Enter, далее то же самое сделаем перед словом «Привет» и перед словом «Твой». В результате письмо будет выглядеть более удобным для чтения. Дальнейшее оформление произведем позже.

Набранное письмо можно напечатать. Для этого нужно «нажать» кнопку Печать (см. рис. 1.2), т. е. установить указатель мыши на изображение кнопки Печать и нажать левую кнопку мыши. Если принтер был включен и настроен, то вы получите копию вашего письма на бумаге. Сделаем это.

Теперь письмо имеется в виде «твердой копии» (на бумаге), а также в виде изображения на экране. Кроме того, хотелось бы на будущее сохранить копию в компьютере, назвав это письмо каким-либо именем, чтобы в последующем по этому имени его можно было найти. Для этого нужно нажать на кнопку Сохранить (см. рис. 1.2), в результате чего откроется окно для работы с файлами (рис. 1.3), вам будет предложено присвоить вашему письму (вашему файлу) имя, например, doc1.doc. Имя файла doc1 означает, что это первый документ, а после точки в имени файла стоит так называемое расширение .doc. Это расширение указывает на происхождение документа именно из текстового редактора Word. На этом этапе не будем придумы-

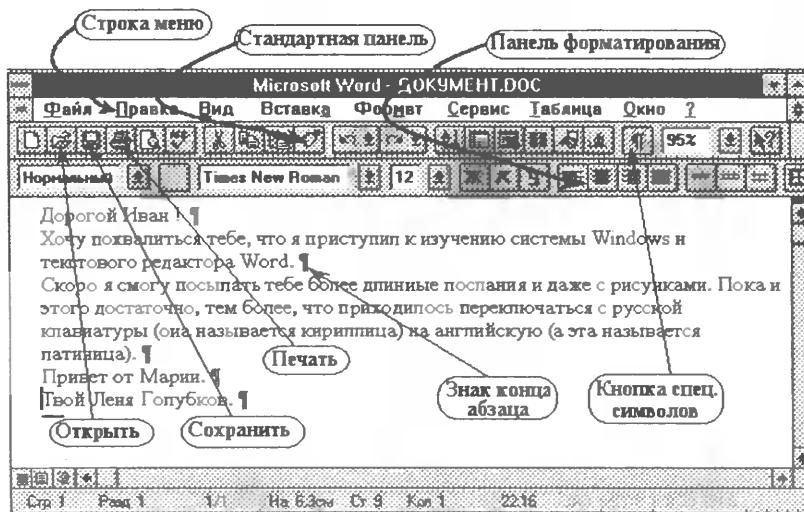


Рис. 1.2

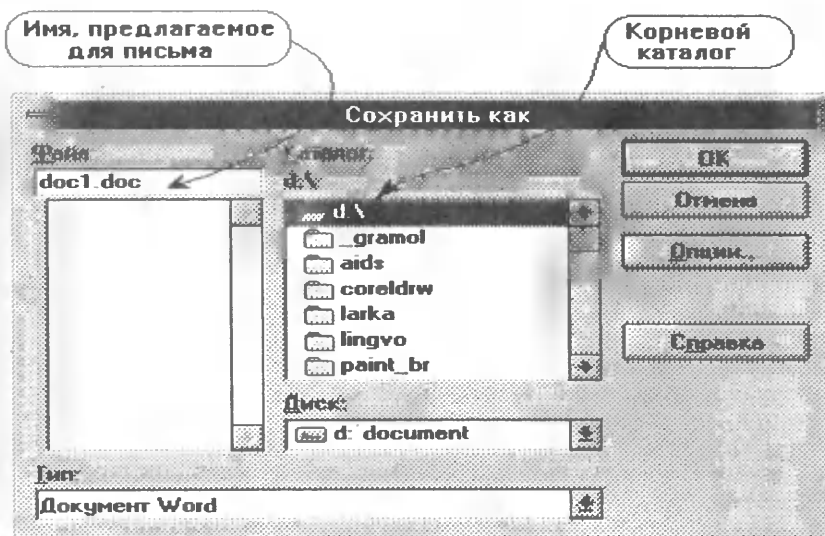


Рис. 1.3

вать свое уникальное имя, а согласимся с предложенным именем, т. е. просто нажмем кнопку **ОК**. Наше письмо, названное теперь именем **doc1.doc**, запишется в некоторый каталог, являющийся хранилищем многих документов. Таких каталогов может быть много, и потому каждый каталог тоже должен иметь свое имя. Имя того каталога, в который мы загружаем наше письмо, приведено в средней части окна под словом **Каталог**. Наш каталог затенен и имеет имя **d:\** (компьютер может быть так настроен, что это окажется каталог с именем **c:\**, это самый общий каталог, называемый **корневым**). Обычно в **корневой** каталог стараются не размещать документы, поскольку при их большом количестве разобраться в них будет довольно затруднительно. Во втором уроке мы введем собственные каталоги, а пока, для простоты, воспользуемся каталогом **d:\**. Позже мы из этого каталога письмо достанем, оформим, переименуем и, если нужно, отредактируем.

Текстовый редактор **Word** после сохранения документа может попросить заполнить «сводку», но на этом этапе мы эту процедуру пропустим, просто еще раз нажав **ОК**. Попутно отметим, что в большинстве случаев нажатие на кнопку **ОК** эквивалентно нажатию клавиши **Enter** на клавиатуре.

Итак, мы имеем письмо на экране, мы получили его распечатку, и, кроме того, оно сохранено в компьютере в каталоге **d:\** под именем **doc1.doc**. Теперь можно это письмо с экрана убрать, т. е. необходимо закрыть файл. В дальнейшем мы широко будем пользоваться термином «**файл**». Согласно определению, **файл** — это поименованная совокупность данных на диске, т. е. это конечное изделие, хранимое в каталоге. В файлах могут храниться письма, документы, таблицы, рисунки и т. д. **Закрытие файла** проведем следующим образом:

- выберем слово **Файл** в верхней строке меню (рис. 1.4) (выбор — это установка указателя мыши на этом слове и последующее нажатие левой клавиши);
- после выбора слова **Файл** появляется ниспадающее меню, из которого необходимо выбрать слово **Закрыть**. При закрытии файла на экране может появиться запрос (рис. 1.5), на который нужно ответить утвердительно, т. е. нажать кнопку **Да**.

Теперь потренируемся в отыскании файла, его открывании (т. е. выводе на экран) и последующем закрывании. Начнем с того, что в стандартной панели нажмем кнопку **Открыть** (рис. 1.2). После появления на экране окна с заголовком **Открыть** (рис. 1.6) приступаем к поиску файла. В среднем окне, под словом **Каталог** находим каталог с именем **d:\** и выбираем его. В результате он будет затенен. Далее с помощью мыши нажимаем на кнопку **ОК**. После этого в левой части окна под словом **Файл** будет выведен список файлов, документов, писем и т. п., хранящихся в этом каталоге. При этом важно проследить, чтобы в окошке **Тип**, расположенном в нижней части

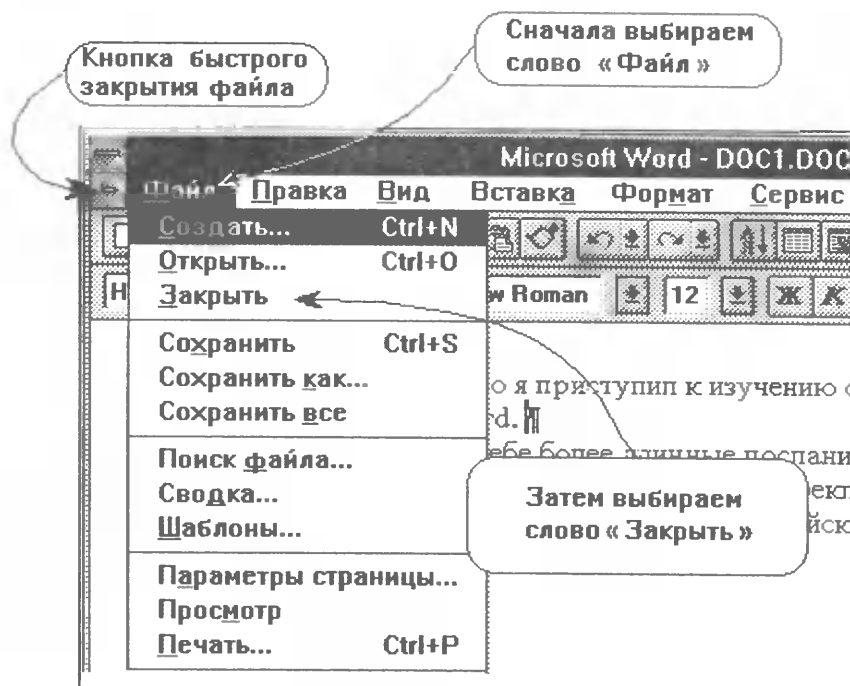


Рис. 1.4

окна, было установлено Документы Word (\*.doc). Найдём в открывшемся списке файлов наше письмо под именем doc1.doc, если имени нашего письма мы не видим, то это значит, что оно находится вне экрана и нужно провести так называемую прокрутку, чтобы оно оказалось в видимой части окна. Для операции «прокрутка» используются стрелки просмотра в правой части подокна (рис. 1.6), при нажатии стрелки направленной вниз в окно поступают имена файлов расположенных ниже просматриваемых каталогов, при нажатии стрелки, направленной вверх, происходит подача файлов сверху.

Выведем содержимое нашего файла (doc1.doc) на экран:

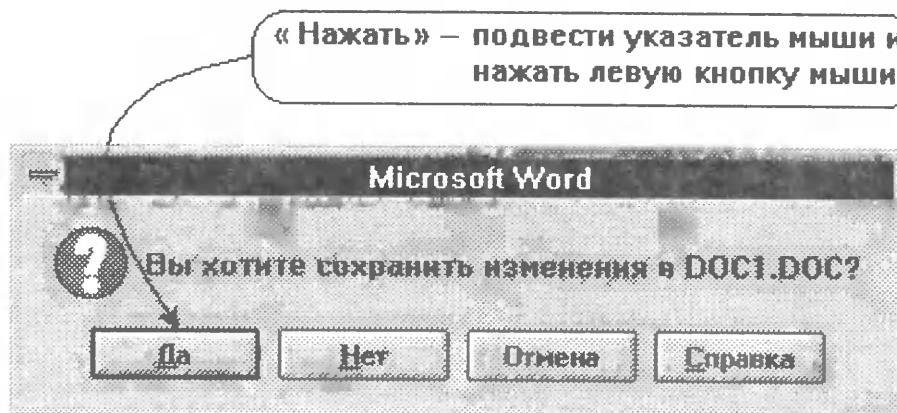


Рис. 1.5

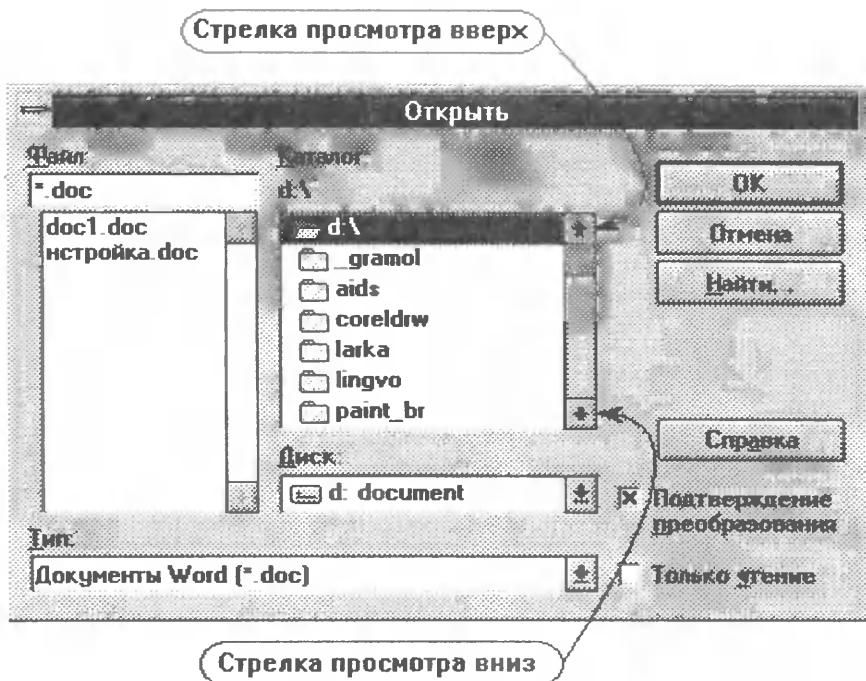


Рис. 1.6

- установим указатель мыши на имени и нажмем левую кнопку, имя файла выделяется;
- «нажмем» на клавишу **ОК**. Через некоторое время перед нами на экране появляется содержимое файла. Ничего не меняя в файле, закроем его;
- выделим из меню слово **Файл**;
- выберем из выпавшего меню слово **Закрывать**.

Теперь еще раз откроем этот файл, но уже более коротким способом. Нажмем кнопку **Открыть**, далее выделим каталог **d:\** и, найдя нужный файл (**doc1.doc**), дважды щелкнем на имени файла — файл открывается без промежуточного нажатия на клавишу **ОК**. Также можно обойтись лишь двойным нажатием на кнопку на экране при закрытии файла. Для этого нужно дважды нажать на кнопку, находящуюся левее слова **Файл** (рис. 1.4).

Большинство окон в Windows закрывается именно таким образом. Для более детального ознакомления с этой процедурой запустите **Учебник по Windows** и задайте демонстрацию и обучение работе с окнами.

Если во время нахождения письма на экране вы сделали некоторые изменения, например добавили дату написания письма, то после закрытия появляется запрос (рис. 1.5), на который нужно ответить **Да**, если вы хотите внесенные изменения сохранить.

В заключение урока попробуйте вновь открыть файл **doc1.doc** одним из двух способов, что-нибудь измените в файле, например поставьте дату, и в завершение закройте файл.

На следующем уроке мы переместим это письмо из общего (корневого) каталога в специально созданную личную папку — каталог.

А на этом этапе завершим работу с Winword и вообще с Windows. Простейший способ выйти из этих программ — нажать на клавиатуре одновременно две клавиши **Alt+F4** (в таких обозначениях знак «+» указывает на одновременное нажатие нескольких клавиш). Для окончательного выхода их, может быть, придется нажать несколько раз и ответить на появляющиеся на экране запросы-уточнения.

## Урок 2. Как создать личную папку (каталог) для хранения писем, а затем поместить туда ранее написанное письмо

На прошлом уроке написанное письмо было сохранено в общем каталоге `d:\`. Однако по мере накопления документов поиск их становится все более затруднителен, даже если им давать запоминающиеся имена. Для удобства хранения и последующего поиска используется система каталогов и подкаталогов, образующих «дерево» каталогов. Таким образом, каталог можно рассматривать как место размещения документов, рисунков, таблиц и т. п., записанных в виде файла. Зная имя файла, можно производить его поиск, вызов и сохранение.

На рис. 2.1 приведен пример дерева каталогов. В основе его применен принцип адресации, широко используемый в обычной почте. Поясним это. Самый верхний каталог под именем **Москва** (на рисунке он выбран) содержит несколько подкаталогов (носящих имена улиц Варварка, Петровка), которые, в свою очередь, являются каталогами, содержащими номера домов, и так далее, вплоть до последнего каталога **ящик1**, в котором находятся разные файлы, в том числе и наше письмо. Если мы поместим письмо в каталог **ящик1**, то при последующем поиске нужно знать, что этот каталог находится в каталоге **стол**, тот в свою очередь, в каталоге **кабинет**, далее в каталоге **кв5**, потом **дом\_27**, затем **Петровка** и наконец каталог **Москва**.

В итоге адрес этого документа можно записать следующим образом:

`d:\Москва\Петровка\дом_27\кв5\кабинет\стол\ящик1\письмо.doc`.

Такую запись иногда называют «путь», имея в виду, что по указанному пути будет отыскиваться адресуемый файл. Начинается путь с указания диска (в нашем случае диск `d`), далее ставится двоеточие, затем идут все имена каталогов, отделяемые друг от друга косой чертой. При написании пути пробелы нигде не ставятся.

Все каталоги и файлы, конечно, удобно именовать по-русски, тем более что русифицированная версия Windows позволяет это делать. Однако проблемы начнутся, если вы будете переносить свои каталоги на другие компьютеры, на которых может быть установлена другая таблица русификации. Поэтому, если вы работаете только



Рис. 2.1

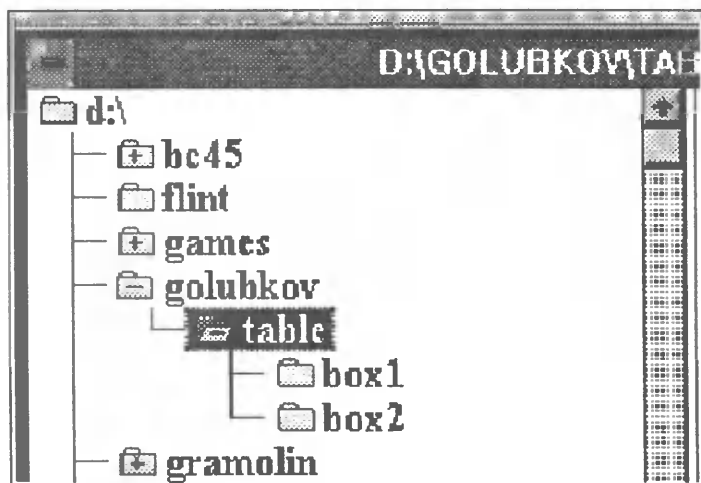


Рис. 2.2

на своем компьютере, можно использовать кириллицу, но в общем случае для имен каталогов и файлов лучше применять латиницу.

Теперь вернемся к задаче построения своего собственного каталога, который будем использовать при размещении всех своих учебных и деловых файлов. Наша цель — создание для собственных нужд дерева каталогов, приведенного на рис. 2.2. На рисунке каталог **golubkov** стоит четвертым в списке каталогов. В каталоге **golubkov** мы расположим каталог **table**, а в нем два каталога: **box1** и **box2**. В одном из этих каталогов, например в каталоге **box1**, будет расположено наше письмо.

Создание новых каталогов, перемещение файлов из одних каталогов в другие, переименование файлов и каталогов — все эти операции удобнее выполнять не в редакторе Winword, а в специальной программе, называемой **Диспетчер Файлов**. Эта программа, так же как и Word, работает под управлением системы Windows. Для открытия **Диспетчера Файлов** необходимо дважды щелкнуть на иконке этой программы. Сама эта иконка обычно находится в папке **Главная** и выглядит так, как это показано (рис. 2.3).

Итак, вы открыли **Диспетчер Файлов** (рис. 2.4), здесь нужно обратить внимание на два момента: список доступных дисков (этот список приведен в верхней части окна **Диспетчера Файлов**) и подокна некоторых из этих дисков. На рис. 2.4 в списке имеются четыре диска (диски **a** и **b** — это обозначения дисководов для гибких дисков, диски **c** и **d** — это жесткие диски компьютера). На этом же рисунке внутри окна **Диспетчера Файлов** открыты два подокна, в которых показано содержимое дисков **c**



Рис. 2.3

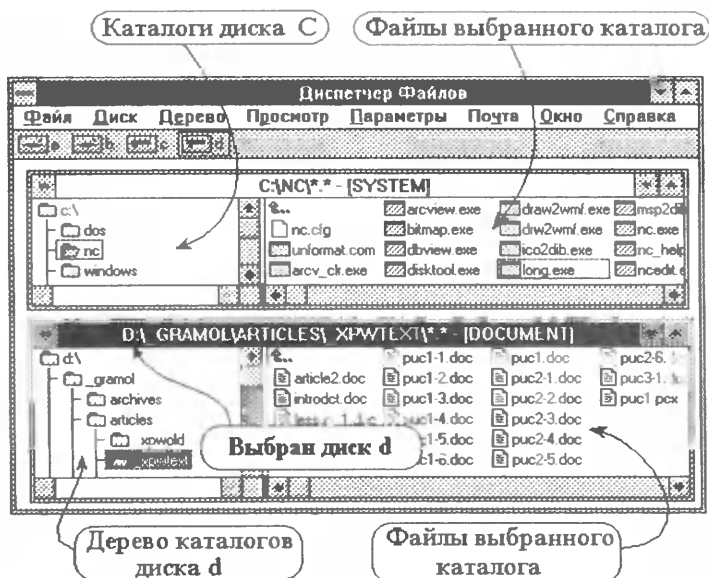


Рис. 2.4

и **d**. Таких подокон может быть много (по количеству дисков), но в каждый данный момент выбранным может быть только одно подокно и только с каталогами этого подокна можно оперировать. Для выбора подокна нужно щелкнуть мышью в пределах этого окна, при этом заголовок окна окажется затенен. В левой панели открывшегося подокна находится дерево каталогов (или часть этого дерева), один из каталогов — выбран (он либо затенен, либо находится в рамочке). В правой панели находится содержимое этого выбранного каталога. Прежде чем начать манипулировать с каталогами и файлами, разберемся, как вызываются диски в Диспетчер Файлов. Для этого нужно знать всего три приема — открытие нового окна, смену диска в открытом окне и закрытие окна.

- Для открытия нового окна с содержимым конкретного диска нужно дважды щелкнуть мышью на пиктограмму с обозначением этого диска.
- Для смены диска необходимо выбрать окно заменяемого диска (щелкнуть один раз в пределах этого окна), затем активизировать пиктограмму с обозначением нового диска.
- Закрытие окна производится (как и во многих остальных случаях при закрытии окон в Windows) с помощью двойного щелчка по прямоугольнику, находящемуся в левом верхнем углу убираемого окна.

Как и с любыми окнами в Windows, с подокнами можно производить различные манипуляции — сдвигать границы, менять местоположение и т. п. Все окна можно модифицировать в размерах, установив курсор на границу окна, и, когда он изменит свой вид на стрелку, нажать кнопку мыши и, не отжимая ее, переместить границу до нужного положения, после чего кнопку отпустить. Для сдвига окна необходимо установить указатель мыши на верхнюю выделенную полосу окна, нажать кнопку мыши и, не отпуская ее, передвинуть мышью, наблюдая за передвижением окна на экране. В тот момент, когда кнопка мыши будет отжата, окно каталогов установится на новом месте. Для освоения и закрепления всех этих манипуляций можно воспользоваться Учебником по Windows.

Гораздо чаще, правда, требуется не просто изменять размеры окон или передвигать их, а так их упорядочить, чтобы они были видны не накладываясь друг друга. Для этой цели удобнее в верхней строке меню выбрать слово **Окно**, а в выпавшем меню выбрать **Горизонтальная мозаика**.

Теперь займемся размещением нашего документа **doc1.doc** в специально создаваемом каталоге. Выполнять это будем в следующем порядке: сначала на диске **c**

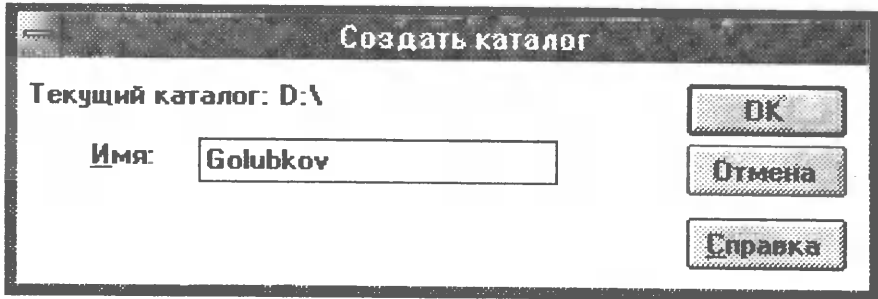


Рис. 2.5

создадим дерево каталогов, далее в один из каталогов этого дерева перенесем наше письмо, а потом переименуем его.

Поскольку наше письмо находится на диске **d**, а мы хотим разместить его в каталогах на диске **c**, необходимо в окне диспетчера файлов разместить два подокна: для диска **d** и диска **c** (рис. 2.4). Создание каталога на диске **c** начнем с того, что активизируем его, далее выберем корневой каталог **c:\**, если в пределах окна этот каталог не виден, нужно прокрутить содержимое окна стрелками. Далее выберем из меню слово **Файл**, потом из открывшегося меню выберем **Создать каталог**, после чего на экране появится окно для ввода имени создаваемого каталога (рис. 2.5). При создании каталога вы, возможно, захотите ввести каталог со своей фамилией. При этом необходимо помнить, что количество букв не должно превышать 8 и в имени каталога не должно быть пробелов, точек и других вспомогательных знаков. Набираем имя **golubkov** или свою фамилию, лучше латинскими буквами, и нажимаем кнопку **ОК**. На экране увидим, что в корневом каталоге появился каталог с введенным именем (рис. 2.6). Теперь необходимо внутри этого каталога создать каталог **table**. Для это нужно щелкнуть мышью на имени нового, только что созданного каталога **golubkov** (или на вашем именованном каталоге) и после того, как он будет выбран (на экране затенен), повторить операцию создания каталога. То есть вначале выбрать слово **Файл**, затем **Создать каталог**, после чего набрать **table** и **ОК**. В вашем собственном каталоге появился подкаталог **table**. И наконец, создадим таким же образом **box1** а затем **box2**. В результате получим дерево каталогов, приведенное на рис. 2.2. .



Рис. 2.6



Следующая задача состоит в переносе ранее написанного письма из корневого каталога **d:\** в каталог **box1**, в котором впредь мы будем хранить письма. Выберем подокно с письмом, установим указатель мыши на имя письма (**doc1.doc**), нажмем кнопку мыши и, не отпуская ее, начинаем перемещать мышшь по подставке так, чтобы появившееся на экране изображение прямоугольника сдвинулось в подокно диска **d** и попало в подкаталог **box1**, теперь кнопку мыши можно отпустить. После этого появится запрос, действительно ли мы хотим копировать файл из каталога Winword диска **c** в каталог **box1** диска **d**. На этот запрос отвечаем **Да**. Дело сделано, но хорошо бы все привести в порядок — убрать файл из диска **c**, так как он там уже не нужен, а перенесенный файл переименовать, чтобы в последующем его можно было легко найти.

Для стирания файла в каталоге диска **c** нужно выбрать подокно диска **c**, т. е. активизировать его, затем выбрать каталог и далее выбрать сам файл. Выделив в меню слово **Файл**, выбираем в выпавшем меню слово **Удалить** и потом в последующем запросе подтверждаем свои действия.

Для переименования файла его нужно выбрать в каталоге **Box1**, затем то же проделать в меню со словом **Файл**, и в открывшемся подменю выбрать слово **Переименовать**. В предложенном окне экрана следует набрать новое имя нашего файла, например **letter1.doc**, что означает письмо номер **1**. Следует еще раз обратить внимание, что в имени файла нет пробела и цифра **1** следует сразу за буквой.

В данном случае можно обойтись без дополнительной процедуры стирания, если использовать не операцию копирования, а перенесения. Перенос отличается от копирования лишь тем, что во время выбора и перенесения файла должна быть нажата клавиша **Shift** (при этом можно заметить, что знак «+» в пределах переносимого прямоугольника пропадает, в отличие от копирования).

Если что-либо в процессе манипуляций с окнами или файлами не получилось — воспользуйтесь кнопкой **Отменить**, а потом пробуйте вновь.

### Урок 3. Как оформить письмо так, чтобы было красиво

В этом заключительном уроке наше письмо, которое уже размещено в нужном каталоге, постараемся оформить так, чтобы оно было красиво и удобно для чтения, т. е. подготовим письмо для печати. Поставим себе задачу превратить текст письма из рисунка 1.2 в текст, показанный на рисунке 3.1. Однако первое, что необходимо сделать, — это проверить письмо на орфографические ошибки и лишь потом можно заняться оформлением. Ошибки можно проверить, просто читая текст, и по мере их обнаружения вносить исправления. Но при такой проверке велика вероятность пропуска ошибки. Автоматическая проверка, которой мы сейчас займемся, существенно упрощает этот процесс.

Прежде чем мы приступим к проверке, кратко опишем, как она проходит. Программа ведет проверку, опираясь на имеющийся в компьютере словарь. Если в словаре отсутствует, например, слово «Голубков», то предлагаются разные варианты действий:

- если это редкое слово и оно написано правильно, то нужно активизировать кнопку **Пропустить**;
- если это слово может повторяться в будущем (в этом или последующих документах), то нужно выбрать кнопку **Добавить**, после чего это слово добавляется во внутренний словарь и при последующих встречах с этим словом программа, сверив правильность его написания, пропустит его и перейдет к проверке следующих слов.

В случае, если обнаружена ошибка, возможны несколько вариантов поведения. Их мы рассмотрим ниже на конкретных примерах при проверке нашего письма.

Поскольку наше письмо, видимо, написано без ошибок, придется для демонстрации программы проверки ввести ошибку в текст самим. Например, в слове «длинные» уберем одну букву «н». Сделать это можно так: поставим курсор между двумя буквами «н» в слове «длинные», а затем нажмем клавишу **Delete** и получим ошибочное слово «длиные».

Итак, приступим к проверке. Выберем письмо на экран и нажмем кнопку стандартной панели с буквами **АБВ** — кнопка **Орфография** (рис. 3.1). На экране

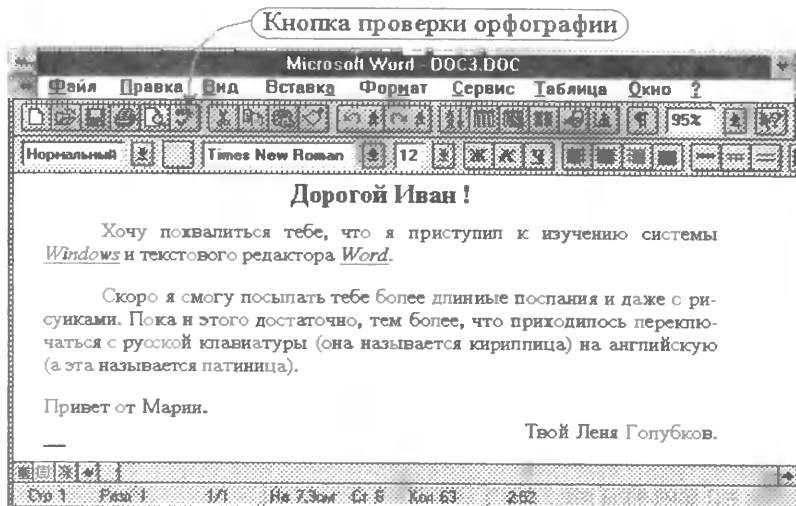


Рис. 3.1

появляется окно проверки орфографии (рис. 3.2), и начинается автоматическая проверка. Если программа показывает слово, которого у нее нет в словаре, но слово написано правильно, то следует выбрать кнопку **Пропустить**. После этого проверка продолжается. Но вот, наконец, очередь доходит до слова, написанного с ошибкой. В окне **Проверка орфографии** либо появляются правильные слова для замены (рис. 3.2), либо таких слов программа предложить не может. Если нужное слово в списке правильных слов есть, например в нашем случае «длинные», то нужно щелкнуть указателем мыши по этому правильному слову и, когда оно будет выбрано, щелкнуть по кнопке **Заменить**. Если же в списке нет нужного нам слова или вообще нет никакого списка, то придется проводить замену вручную, а именно: установить курсор на ошибочное слово в тексте и исправить ошибку. После этого, щелкнув по кнопке **Начать**, проверку можно продолжить. В конце проверки на экране появится сообщение о завершении проверки. Нажав **ОК**, мы вновь возвращаемся к письму, в котором уже нет ошибок, по крайней мере вероятность их появления уменьшилась.

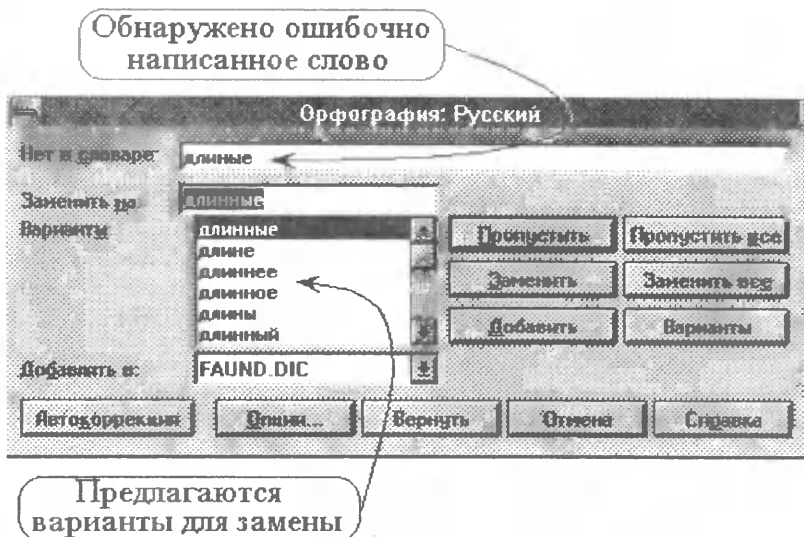


Рис. 3.2

Теперь займемся оформлением письма и начнем со шрифта. Изменить шрифт можно следующим образом: изменить тип и размер шрифта, сделать буквы полужирного или курсивного начертания или подчеркнутыми. Название текущего типа шрифта показано в окне (рис. 3.3). Для изменения шрифта нужно нажать кнопку разворота окна ↓ (рис. 3.3), в выпавшем меню будет показан список шрифтов, подключенных в данный момент к текстовому редактору. Если список подключенных шрифтов весь не вмещается, то следует прокрутить его вниз или вверх, используя стрелки в правой части окна. Правее расположена кнопка выбора размера шрифта и далее кнопки выбора полужирного написания, а также курсивного и подчеркнутого. Для того чтобы воспользоваться всеми этими возможностями, необходимо сначала выделить ту часть текста, к которой мы хотим применить те или иные перечисленные модификации, после чего нажать соответствующую кнопку. Итак, начнем, *step by step*, оформлять письмо до приведения его к виду рис. 3.1.

Шрифт оставим такой, каким письмо было набрано с самого начала, но увеличим размер букв. Сначала выделим весь текст письма. Для этого необходимо:

- установить указатель мыши в левой части экрана так, чтобы он принял вид стрелки, направленной в верхний правый угол;
- одновременно нажать клавишу Ctrl и левую кнопку мыши.

Документ выделен, причем мы выделили его простейшим, но не единственным способом. Как и в большинстве операций в Winword, выделение и другие операции можно производить несколькими способами.

Теперь изменим размер букв:

- щелкните по кнопке ↓, находящейся правее окна с цифрами размера шрифта;
- в выпавшем окне выберите цифру 14 (или, если хотите, цифру 16), щелкнув указателем мыши на этой цифре.

В результате шрифт всего документа изменится и будет произведена автоматическая верстка (при этом увеличится количество строк в документе).

Если больше никаких преобразований мы делать не собираемся, то необходимо снять выделение, щелкнув мышью в любом месте экрана вне выделенного текста. Если оказалось, что получилось не то, что хотелось, то всегда можно вернуться назад, используя для этого кнопку **Отменить** (рис. 3.3). Возврат назад может быть несколько, т. е. можно отменить не только последнее свое действие, но и два, и три, и больше количество последних действий. В процессе возвратов можно отменить возврат, нажав кнопку **Вернуть** (рис. 3.3).

Продолжим модификации письма и приведение его к виду на рис. 3.1. Изменим

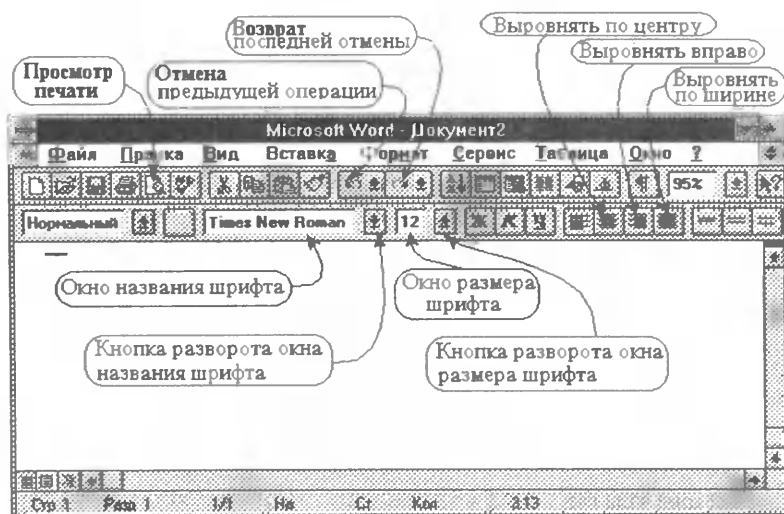


Рис. 3.3

первую строку письма, написав ее полужирными буквами, кроме того, расположим ее в центре листа. Сначала выделим эту первую строку. Для этого нужно установить курсор мыши левее этой строки и, когда указатель мыши примет форму стрелки направленной вверх и влево, нажать кнопку мыши. Появившееся затемнение этой строки указывает на то, что она выбрана. Выбрать эту строку можно иначе — просто трижды щелкнув указателем мыши по этой строке. Этот прием используется для выделения целого абзаца. Выделив строку и подготовив ее тем самым к модификациям, приступим к ним. Сначала увеличим шрифт так, как это уже делали ранее. В опустившемся меню размеров шрифта выберем для первой строки размер шрифта 18. Далее, не снимая выделения, нажмем на кнопку Ж (рис. 3.3), сделав тем самым текст этой строки полужирным и, наконец, расположим эту строку по центру, для чего нажмем на кнопку **Выровнять по центру** (рис. 3.3). Теперь можно снять выделение, щелкнув указателем мыши в любом месте вне выделенной строки. Здесь также можно потренироваться в отмене модификаций и в возврате обратно с помощью кнопок **Отменить** и **Вернуть**.

Следующую модификацию проведем над отдельным словом. Изменим написание слова Windows и напишем его курсивом. Сначала выделим это слово, дважды щелкнув по нему указателем мыши, а затем нажмем кнопку К и потом Ч (рис. 3.3), после чего это слово окажется написанным курсивом и подчеркнутым. Прodelайте аналогичную операцию со словом Word.

Текст будет смотреться более красиво, если каждый абзац будет начинаться с красной строки. Выполним это оформление. Как всегда, сначала нужно выделить весь модифицируемый текст. Если форматировать только один абзац, то нужно внутри абзаца трижды щелкнуть указателем мыши. Попробуйте это. В нашем случае нужно выделить сразу два абзаца от слова «Хочу» до слова «латиница», чтобы не форматировать каждый абзац по отдельности. Для выделения произвольного фрагмента текста (в том числе и для двух абзацев) воспользуемся следующим порядком действий:

- установим указатель мыши перед началом первого абзаца и щелкнем кнопкой мыши — при этом курсор экрана установится в начале первого абзаца;
- установим указатель мыши в конец второго абзаца (иногда для этого нужно прокрутить экран стрелками прокрутки так, чтобы нужный конец абзаца оказался в видимой части экрана), нажмем клавишу Shift и, не отпуская ее, щелкнем клавишей мыши в конце выделяемого участка текста. Будет выделен весь текст между первой и второй установками курсора.

Другой вариант выделения произвольного фрагмента текста требует уверенного навыка обращения с мышью. Навык состоит в том, что указатель мыши ставится перед началом выделяемого текста, нажимается и держится нажатой кнопка мыши, далее мышь с нажатой кнопкой перемещается по подставке и в процессе движения выделяется текст. После выделения нужной части текста кнопка мыши отпускается, а на экране остается выделенный текст.

Для начинающих мы рекомендуем воспользоваться первым вариантом выделения текста, не требующим особых навыков обращения с мышью.

Итак, два абзаца выделены, и теперь можно заняться их форматированием. Выбираем в строке меню слово **Формат**, а в выпавшем меню слово **Абзац**, после чего попадаем в окно форматирования абзаца (рис. 3.4). Затем активизируем стрелку под словами **Первая строка** и в выпавшем меню выделяем слово **Красная**. Прежде чем дать команду на выполнение, т. е. прежде чем нажать на **ОК**, подготовим еще один элемент форматирования, а именно установим между абзацами интервал. Для этого в том же окне форматирования абзаца в части **Интервалы** в подокне **После** выберем некоторый интервал, который будет установлен после абзаца. Выбор интервала осуществляется нажатием стрелок рядом со словом **После**. Установим интервал в 12 пт (пунктов). Далее нажмем **ОК** и тем завершим форматирование абзацев. В абзацах должна появиться красная строка и установлен интервал.

Хорошо бы еще установить интервал между первой строкой и остальным текстом. Выполним это, используя более простой способ: установим курсор в конце первой строки и нажмем клавишу Enter. Теперь сместим подпись под письмом вправо. Для этого нужно выделить последнюю строку, а затем нажать кнопку **Выровнять вправо** (рис. 3.3). На этом форматирование отдельных частей текста завершим и теперь



Рис. 3.4

посмотрим, как письмо выглядит в целом. С этой целью в стандартной панели нужно нажать соответствующую кнопку с изображением лупы **Просмотр печати** (рис. 3.3).

При просмотре можно видеть, что правое поле письма довольно неровное по сравнению с левым, кроме того, все письмо расположилось в верхней части листа. Внесем последний штрих в оформление нашего документа — подравняем поля и сделаем их шире, чтобы текст заполнил лист более равномерно.

Для выравнивания правого края текста введем переносы слов. Для этого нужно выбрать в строке меню слово **Сервис**, а в выпавшем меню опцию **Перенос слов**. В появившемся окне **Перенос слов** нужно нажать на **Автоматический перенос слов в документе** и далее на кнопку **ОК**. Письмо теперь имеет более выровненное правое поле. Можно сделать так, чтобы правое поле было таким же ровным, как и левое, для этого нужно выделить два абзаца текста и нажать кнопку **Выровнять по ширине** (рис. 3.3).

А теперь для более ровного расположения письма на листе увеличим поля у листа слева, справа и сверху. Последовательность действий такова: выбираем из строки меню слово **Файл**, в выпавшем меню выбираем **Параметры страницы** и попадаем в окно установки параметров страницы (рис. 3.5). Увеличим последовательно левое, правое и верхнее поля до 4 см. Увеличение происходит нажатием соответствующих стрелок рядом с названиями полей. Завершая форматирование, щелкнем по кнопке **ОК**. Вот теперь письмо оформлено окончательно и выглядит так, как показано на рис. 3.1.

В заключение распечатаем это красивое письмо. Можно было бы просто нажать кнопку **Печать**, но давайте распечатаем это письмо в двух экземплярах. Для этого нужно выбрать в меню слово **Файл**, затем в выпавшем меню слово **Печать**. В открывшемся окне **Печать** можно заказать печатать некоторое количество копий. Установка количества копий выполняется по стрелке рядом со словом **Копий**. Установим цифру 2 и нажмем **ОК**.

После получения распечатки, полюбовавшись ею, закроем файл. Произведем двойное нажатие на левом верхнем квадрате и через некоторое время на экране возникнет запрос о том, хотим ли мы сохранить все наши труды по форматированию. Мы, конечно, хотим и потому выбираем кнопку **Да**.

Одну копию письма оставим себе, другую отправим Ивану по почте. Хотя, конечно, пора уже отправлять документы по факсу прямо из компьютера. Это экономит время, бумагу и вообще очень удобно. Но это предмет наших будущих уроков. Будем мы изучать и построение таких таблиц, которые приведены ниже, и многое другое.

Некоторые поля изменили для равномерного заполнения листа

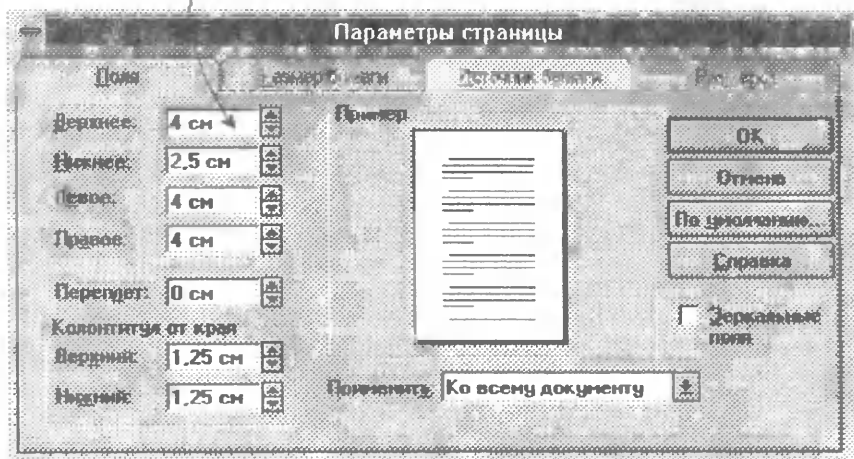


Рис. 3.5

В заключение приведем три небольшие таблицы для использования на первых порах при работе в Winword.

**Таблица 1. Установка курсора в разные части текста**

| Место установки      | Нажать клавиши |
|----------------------|----------------|
| в начало документа   | Ctrl + Home    |
| в конец документа    | Ctrl + End     |
| в начало строки      | Home           |
| в конец строки       | End            |
| на одно слово вправо | Ctrl + →       |
| на одно слово влево  | Ctrl + ←       |

**Таблица 2. Выделение фрагмента текста**

| Выделяемый фрагмент     | Действия  |
|-------------------------|---|
| слово                   | два щелчка на слове   |
| абзац                   | три щелчка на абзаце  |
| строка                  | установить указатель мыши в левой части экрана, пока он не приобретет вид стрелки, направленной в правый верхний угол, далее сделать щелчок левой кнопкой мыши                            |
| столбец                 | Alt + мышь  |
| весь документ           | установить указатель мыши в левой части экрана, пока он не приобретет вид стрелки направленной в правый верхний угол, далее при нажатой клавише Ctrl произвести щелчок левой кнопкой мыши |
| произвольный фрагмент   | 1-й способ: установить курсор на месте начала фрагмента, далее, нажав клавишу Shift, щелкнуть клавишей мыши в конце выделяемого фрагмента   |
|                         | 2-й способ: установив курсор на месте начала фрагмента, нажать левую кнопку мыши и, не отпуская ее, продвигать мышь до тех, пор пока не будет выделен нужный фрагмент                     |
|                         | 3-й способ: нажатая клавиша Shift + клавиши ← ↑ → ↓ (этим вариантом удобно пользоваться при выделении части слова)  |
| снятие любого выделения | щелкнуть клавишей мыши в любом месте вне выделенного фрагмента  |

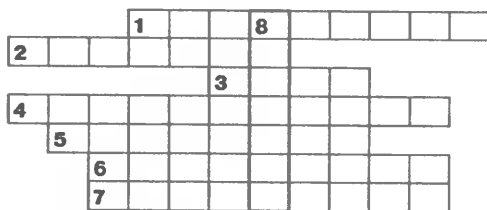
**Таблица 3. Перемещение или копирование фрагментов текста в пределах видимости на экране**

| Перемещение   | Копирование  |
|---|--|
| <p>Для перемещения уже выделенного фрагмента нужно:</p> <ul style="list-style-type: none"> <li>• поместить указатель мыши внутрь выделенного фрагмента, после чего он приобретает вид стрелки, направленной в левый верхний угол;</li> <li>• нажать левую кнопку мыши и через некоторое время рядом с указателем появится прямоугольник и вертикальная полоска;</li> <li>• не отпуская кнопку мыши, передвинуть и установить эту полоску в то место, куда нужно перенести фрагмент текста;</li> <li>• отпустить кнопку</li> </ul> | <p>Для копирования выделенного фрагмента нужно:</p> <ul style="list-style-type: none"> <li>• проделать то же самое, что и при перемещении, но перед отпусканием кнопки мыши нужно нажать клавишу Ctrl, при этом рядом с вертикальной полоской появится знак +, что означает добавление в текст копируемого фрагмента.</li> </ul> |

\* \* \*

## КРОССВОРД

Составитель — Г. Г. Заковряжина (г. Нягань, Тюменская обл.)



**По горизонтали:**

1. Устройство для автоматической обработки информации.
2. Внешнее устройство ЭВМ для отображения графической и текстовой информации, в котором изображение строится посредством электронно-лучевой трубки.
3. Вид внешней памяти ЭВМ, устройство для хранения информации.
4. Объем информации, обрабатываемой ЭВМ за одну операцию.
5. Внешнее устройство ЭВМ, используемое для ввода информации, позволяющее пользователю вручную перемещать курсор по экрану дисплея.
6. Устройство для обработки информации в ЭВМ.
7. Упорядоченная последовательность действий для ЭВМ, реализующая алгоритм решения задачи.

**По вертикали:**

8. Печатающее устройство.

**С. В. Русаков,**

*доктор физико-математических наук, Пермский государственный университет,  
Лаборатория информатизации образования Пермской области*

**И. Г. Семакин,**

*кандидат физико-математических наук, Пермский государственный университет,  
Лаборатория информатизации образования Пермской области*

## БАЗОВЫЙ КУРС ОИВТ: «ПЕРМСКАЯ ВЕРСИЯ» (часть 7)\*

### Тема 10. Компьютер — связист

Вспомним схему «Человек и информация» из статьи [1]. Важнейшим предметом интеллектуальной деятельности человека является обмен информацией. Также мы говорили, что процесс информационного обмена между узлами компьютера происходит непрерывно. Обмен информацией между пользователем и машиной осуществляется через клавиатуру, дисплей, принтер. В данной статье речь пойдет о новой форме: обмене информацией между компьютерами, о компьютерных системах телекоммуникаций.

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

В последнее десятилетие к ним прибавилось еще одно средство — это компьютерные коммуникации. Они обещают потеснить факсимильную и телетайпную связь подобно тому, как последние вытесняют сегодня телеграф.

В настоящее время наиболее распространенным способом компьютерных коммуникаций является электронная почта. Связь абонентов электронной почты осуществляется через компьютерные сети [2—4].

Сеть состоит из множества связанных между собой компьютеров. Пользователи сети называются ее абонентами. Абоненты подключаются к центральной ЭВМ, называемой хост-компьютером (от

англ. host — «хозяин») или головной машиной сети.

Хост-машины, в свою очередь, можно соединить, образовав разветвленную коммуникационную систему, состоящую из нескольких связанных между собой узлов. Каждый узел обслуживает работающих в нем пользователей и обеспечивает обмен информацией с другими узлами (см. рис. 1).

В качестве головной машины используют ЭВМ высокой производительности с большим объемом памяти на жестком диске. Хост-компьютер, как правило, обеспечивает одновременную работу нескольких абонентов. Для этого к нему подсоединяют линии связи. Головная машина постоянно включена и готова отвечать на вызовы абонентов сети.

В компьютерных сетях каждый абонент может использовать различные марки компьютеров, типы модемов, линии связи, коммуникационные программы. Чтобы все это оборудование работало согласованно, работа сетей подчиняется специальным техническим соглашениям, которые называются протоколами.

Протоколы — это стандарты, определяющие формы представления и способы пересылки сообщений, процедуры их интерпретации, правила совместной работы различного оборудования в сетях.

Международная организация по стандартизации (ISO) подготовила и ввела в действие многоуровневую (иерархическую) структуру протоколов.

Работу головной машины обеспечивает специальная сетевая программа, которая ведет диалог с пользователями и поддерживает все действующую

\* Продолжение. Начало см.: Информатика и образование. 1994. № 5, 6; 1995. № 1, 2, 3.



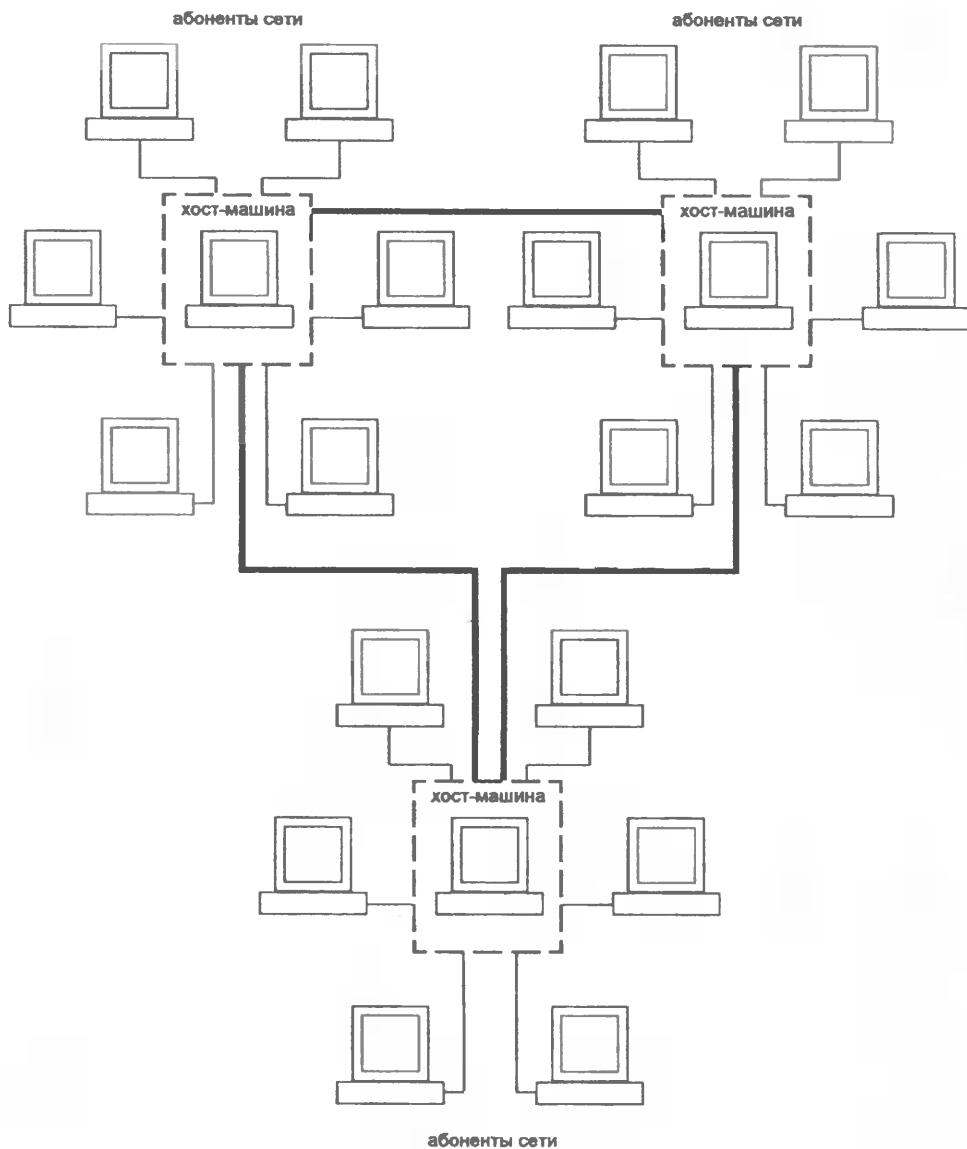


Рис. 1

щие в сети протоколы связи. Сегодня в мире используются десятки сетевых программ, имеющих различный пользовательский интерфейс. Поэтому в каждой сети надо осваивать принятые здесь технические правила работы, соглашения о способах адресации корреспонденции и т. п.

Наиболее популярным видом услуг, которые предоставляют своим абонентам

компьютерные сети, является электронная почта (E-mail).

Каждый абонент электронной почты при регистрации получает свой собственный «почтовый ящик» — некоторый объем памяти на хост-машине, в который попадают все адресованные ему сообщения. Имена почтовых ящиков и сведения об их владельцах доступны всем абонентам сети. Войдя в сеть, можно послать сообщение по любому адресу,



Рис. 2

и это сообщение попадет в соответствующий почтовый ящик. Для получения поступившего письма необходимо сообщить системе имя почтового ящика и пароль, дающий право на получение информации. Каждый владелец почтового ящика сам устанавливает пароль, закрывая тем самым свой почтовый ящик от посторонних.

Используя различные протоколы, можно положить в почтовый ящик своего корреспондента как текстовые, так и двоичные файлы. С помощью двоичных файлов передают любые виды данных: программы для ЭВМ в машинных кодах, звук, изображение, текстовые файлы с любой кодировкой символов. Единственное требование состоит в том, чтобы адресат сумел правильно проинтерпретировать (понять) переданное ему сообщение.

Главным ограничением при передаче данных по сети является объем передаваемых данных. Страница машинописного текста имеет объем около 2,5 Кб. Картинка размером со страницу может потребовать файл в десятки раз большего объема. Зная скорость передачи данных по компьютерной сети, можно легко подсчитать, сколько времени займет пересылка одной картинке по сети.

Вне зависимости от того, кем и как организована сеть, все абоненты используют на своих рабочих местах компьютерные коммуникационные системы, состоящие, как правило, из одних и тех же компонентов. Состав рабочего места абонента изображен на рис. 2.

При изучении данной темы будем рассматривать рабочее место абонента как исполнителя, назначение которого — информационный обмен в рамках компьютерной сети. Назовем его исполнителем-связистом.

### Аппаратные средства

Абоненту телекоммуникационной сети нужны компьютер с соответствующей программой, модем и линия связи,

позволяющая компьютеру соединяться с другой компьютерной системой (см. рис. 2). Если воспользоваться общепринятой терминологией, получим следующий набор аппаратных средств.

**Терминал.** Обычно это персональная ЭВМ, используемая для получения и отправки корреспонденции.

**Модем.** Для того чтобы соединить персональный компьютер с телефонной сетью, необходимо специальное устройство, согласующее их работу. Таким устройством является модем (сокращение от слов *модулятор/демодулятор*). Модем переводит двоичные сигналы, используемые ЭВМ, в аналоговые, которые характерны для существующих телефонных линий (работает как демодулятор). Для соединения модема с ЭВМ используется стандартный последовательный порт связи, который имеется у каждого компьютера.

Модем может выполняться в виде отдельного устройства, подключаемого к ЭВМ. В последние годы все чаще используются встроенные модемы в виде электронной платы, устанавливаемой в компьютере.

Одной из важнейших характеристик модема является скорость передачи данных. Сегодня применяются модемы, передающие по телефонной сети данные со скоростью 1200—2400 бод (бод=бит/с) и выше (современные высокоскоростные модемы имеют быстроедействие до 28800 бод).

Пусть используемый модем во время работы в сети может переслать 1200 бод (или 150 символов в секунду), тогда пересылка полной страницы текста (около 2500 знаков) займет около 17 с. Переключение модема на 2400 бод удвоит скорость передачи. Модем, имеющий высокую скорость, как правило, позволяет работать с низкой скоростью.

Максимальную скорость передачи данных ограничивает и качество телефонной сети. Лучшие из сегодняшних модемов в состоянии передавать информацию даже по недостаточно качествен-

ным телефонным линиям со скоростью 1200 бод. Для этого они имеют специальные средства корректировки ошибок, возникающих в процессе передачи данных. Такие модемы сравнительно дороги, но они обеспечивают связь практически через любую телефонную сеть и выполняют множество вспомогательных функций. Эти модемы часто называют «интеллектуальными».

Линия связи. Для компьютерной коммуникации используют коммутируемые телефонные линии, выделенные линии связи и каналы цифровой связи.

Еще не так давно выделенные линии были основными каналами связи между ЭВМ на расстоянии. Они достаточно надежны, а скорость передачи данных по ним составляет от нескольких тысяч до нескольких десятков тысяч бод. Однако выделенные линии дороги и, как правило, не очень экономичны. Сегодня они используются в системах, где постоянно передаются большие объемы информации или требуется срочная надежная передача данных (системы резервирования билетов и т. п.).

С появлением высококачественных модемов коммутируемые телефонные линии становятся наиболее распространенными каналами связи для компьютерной коммуникации. Когда модем не работает, можно использовать эту линию для обычного телефонного разговора, так как телефонный аппарат обычно подключен в телефонную сеть параллельно с модемом. Телефонные линии широко распространены, сравнительно дешевы и доступны практически каждому. В то же время коммутируемые телефонные каналы сильно зашумлены, а скорость передачи данных по ним ограничена и составляет от нескольких сотен до нескольких тысяч бод. Появление персональных компьютеров и «интеллектуальных» модемов, способных работать с коммутируемыми телефонными линиями, положило начало массовому проникновению компьютерной коммуникации в сферу управления, в научно-исследовательские учреждения, на производство, в быт, в сферу образования.

Пропускная способность каналов цифровой связи составляет от десятков тысяч до сотен миллионов килобод. Они используются для быстрой передачи между ЭВМ больших и очень больших объемов информации. Развитие цифровых каналов приводит к революции в

технике связи, открывает огромные возможности для абонентов компьютерных сетей, обещает интегрировать в единое целое все существующие сегодня средства связи.

Программное обеспечение электронной почты — коммуникационная программа (КП). Используемое в настоящее время ПО такого рода достаточно разнообразно (характеристики некоторых из них см. в табл. 1). В качестве примера рассмотрим некий обобщенный «образ» такой программы.

### Среда КП

Рабочим полем КП является экран дисплея, на котором располагаются меню, строка состояний, рабочие окна. В рабочие окна вызываются списки писем и сами письма (сообщения).

Важнейшим элементом среды КП является почтовый ящик — область внешней памяти хост-машины, куда поступают письма. Во внешней памяти терминала для долговременного хранения писем используется почтовый архив, а для адресов постоянных абонентов — адресный справочник.

Электронное письмо обычно состоит из краткого сообщения (текста) и пересылаемых файлов, в которых может содержаться самая разнообразная информация: тексты, программные продукты, закодированные графические изображения и т. д. Для хранения этих файлов создаются специальные каталоги почты.

Как и всякий программный продукт с дружественным интерфейсом, КП имеет справочник-подсказку.

На рис. 3 схематически показано размещение программных и информационных средств КП в памяти хост-машины и терминала.

### Режимы работы КП

Установка параметров конфигурации. Этот режим чаще всего используется в начальный момент подключения абонента к электронной почте. При этом создаются почтовый ящик, почтовый архив, адресный справочник, каталоги почты. Кроме того, устанавливаются номер телефона и параметры модема, к которым подключается терминал.

В процессе текущей работы в этом режиме можно управлять планировани-

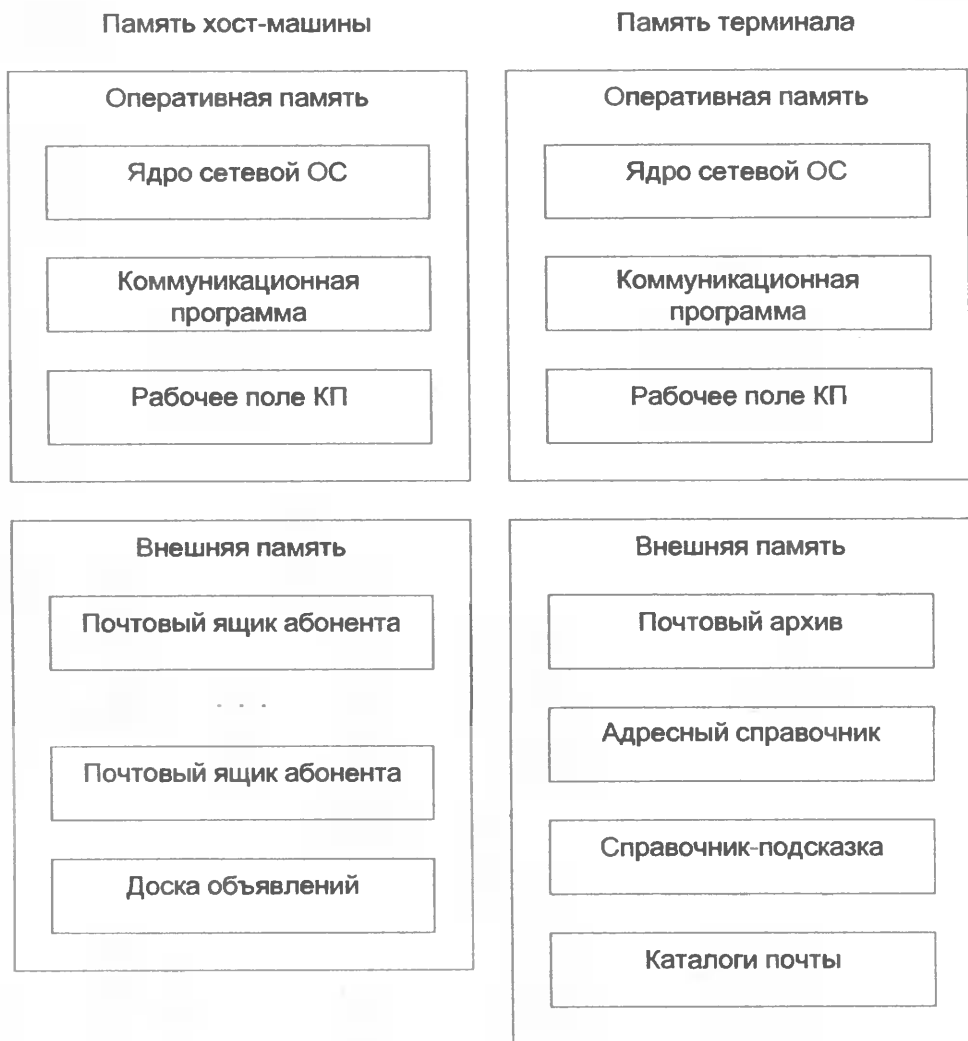


Рис. 3

ем времени обмена почтой и устанавливать текущего активного пользователя (если к данному терминалу имеют доступ несколько абонентов).

**Просмотр почтового ящика (списка писем).** Во время просмотра можно отсортировать полученные письма (например, по дате отправления, по имени отправителя и т. д.) и выбрать письмо для просмотра.

**Просмотр письма.** В этом режиме помимо визуального просмотра письма можно выполнять следующие действия над письмами:

- удаление;
- сохранение в почтовом архиве;
- переписывание в файл;
- пересылка другому адресату;
- печать на принтере.

**Подготовка/редактирование писем.** Письмо подготавливается в специальном рабочем поле — бланке письма, который содержит адресную часть, место для краткого описания (сути) письма, место для указания имен файлов, отправляемых с этим письмом. Для работы на этом бланке используется встроенный текстовый редактор. Заполнение ад-

ресной части можно осуществить выбором из списка адресов. Имена отправляемых файлов можно выбрать из каталога почты.

Отправление электронной корреспонденции. В этом режиме подготовленное письмо отправляется адресату. При этом можно использовать дополнительные услуги, например уведомление о получении.

### Система команд КП

Все команды можно условно разделить на две группы: переключение (переход) режимов работы и команды, обслуживающие конкретный режим работы. Перечень последней группы команд отчетливо просматривается в описании режимов работы КП.

### Данные

Коммуникационная программа имеет дело с данными трех видов:

- адрес получателя письма;
- краткое сообщение;
- пересылаемые файлы.

### Дополнительные виды услуг для абонентов компьютерных сетей

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

- доступ к базам данных;
- электронные доски объявлений;
- информационные услуги.

**База данных.** Доступ к базам данных — типичный вид услуг, предоставляемых абонентам компьютерной сети. Подключившись к сети через телефонную линию и задав сетевой адрес нужной ему базы данных, абонент подключается к ней и в режиме диалога может получить нужную ему информацию. Стоимость информационных услуг

обычно прямо пропорциональна времени работы с системой.

**Электронные доски.** Электронная доска объявлений по своему замыслу аналогична обычной доске объявлений, которая висит на стене в каждой школе. При использовании базы данных любой абонент может прочитать всю хранящуюся там информацию, но не имеет права ее изменять. Пользуясь электронной почтой, абонент может записать информацию в любой почтовый ящик, но прочитать и изменить данные он может только в своем собственном. Электронная доска объявлений позволяет каждому записать любую информацию, представляющую интерес для абонентов системы, и прочитать сообщения, помещенные туда другими абонентами. Такой режим работы дает возможность использовать электронные доски объявлений для проведения компьютерных конференций (телеконференций).

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

Передача сообщения для телеконференции идет в обычном режиме электронной почты, только адресатом в этом случае является конкретная тема в телеконференции.

**Информационные услуги.** Большинство существующих сегодня коммерческих компьютерных сетей предоставляет своим абонентам все перечисленные выше виды коммуникационных услуг, а также разнообразные информационные услуги. Вы можете получать и посылать почту, просматривать сообщения о последних новостях в области спорта и политики, знакомиться с прогнозом погоды, биржевыми сводками, различными электронными изданиями (электронное издание журнала «PC World» выходит на месяц раньше, чем соответствующее печатное издание) и рекламными объявлениями, работать с базами данных, делать заказы на приобретение различных товаров по каталогу. Многие системы предоставляют своим абонентам возможность играть в компьютерные игры.

## Поурочное планирование

### Урок 1.

#### Теоретическая часть.

Вводится понятие телекоммуникации, глобальных компьютерных сетей (в сравнении с локальной сетью КУВТ). Основные компоненты компьютерной сети (хост-машина, линия связи, модем, терминал, коммуникационная программа). Принципы действия электронной почты. Терминал — «Исполнитель-связист».

### Урок 2.

#### Практическая часть.

Если школа имеет выход в какую-нибудь глобальную сеть, провести показательное занятие на терминале этой сети, отработав все режимы электронной почты. В противном случае воспользоваться почтой локальной сети КУВТ, подчеркивая ее сходство и различие с реальной электронной почтой (см. табл. 1).

## Контрольные вопросы

1. Какие основные компоненты (аппаратные и программные) входят в средства компьютерных коммуникаций?
2. В чем отличие терминала от хост-машины?
3. По каким трем основным характеристикам отличаются друг от друга модемы различной конструкции?
4. Сколько символов текста можно передать за 5 с по каналу связи с быстродействием 1200 бод?
5. Какие виды линий связи используются в телекоммуникациях?
6. Какие основные виды услуг представляются абонентам компьютерных сетей?
7. Как работает электронная почта?
8. Как проводятся компьютерные конференции?
9. Сравнительная характеристика компьютерных телекоммуникаций с традиционными видами связи (почтой, телеграфом, телефоном и т. д.).

Т а б л и ц а

| Функциональные возможности E-mail           | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1. Интерфейс                                |   |   |   |   |
| — меню                                      | + | + | + | + |
| — справка                                   | + | + | + | + |
| 2. Установка параметров конфигурации        |   |   |   |   |
| — адресный справочник                       | + | + | — | + |
| — почтовый ящик                             | + | + | + | + |
| — архив писем                               | — | + | — | + |
| — установка текущего активного пользователя | + | + | + | + |
| — планирование времени обмена почтой        | + | + | — | — |
| — установка номера телефона                 | + | + | — | — |
| — установка параметров модема               | + | + | — | — |
| — установка каталогов для почты             | + | + | + | — |
| 3. Работа с почтой                          |   |   |   |   |
| а) просмотр почтового ящика                 |   |   |   |   |
| — сортировка списка писем                   | + | + | + | — |
| — выбор письма                              | + | + | + | + |
| б) просмотр письма                          |   |   |   |   |
| — удаление письма                           | + | + | + | — |
| — сохранение в почтовом архиве              | — | + | — | + |
| — сохранение в виде файла                   | + | + | + | — |
| — печать на принтере                        | + | + | + | — |
| — пересылка другому адресату                | + | + | + | — |
| 4. Подготовка/редактирование писем          |   |   |   |   |
| — заполнение адресной части                 | + | + | + | + |
| — краткое описание сообщения                | + | + | + | + |
| — указание пересылаемых файлов              | + | + | + | — |
| 5. Прием/передача сообщений                 |   |   |   |   |
| — через почтовый ящик получателя            | — | — | + | — |
| — посредством сеанса связи                  | + | + | — | + |

Обозначения:

1 — COMMANDER MAIL

2 — программное обеспечение РЕЛКОМ

3 — почта в локальной сети NetWare, программы PostMan и Mail

4 — почта в локальной сети УКНЦ; программа реализована в Пермском государственном педагогическом университете

IBM PC

IBM PC

УКНЦ

## Литература

1. Семакин И. Г. Базовый курс ОИВТ: «Пермская версия» (часть 2)//Информатика и образование. 1994. № 5.
2. Уваров А. Ю. Компьютерные коммуникации//Информатика и образование. 1991. № 1.
3. Уплинский Е. В., Аввакумов Э. Г. Те-

лекоммуникационная сеть ТТТ-ВГТ//Информатика и образование. 1994. № 4, 6.

4. Мегвегов О. Б., Уваров Ю. А. Региональная образовательная компьютерная сеть//Информатика и образование. 1994. № 5.

**Ю. А. Шафрин,**

*ведущий консультант компьютерного отдела издательства «АВФ»*

## **ЧЕМУ УЧИТЬ ШКОЛЬНИКОВ НА УРОКАХ ИНФОРМАТИКИ?**

Издательство «АВФ» готовит к выпуску пробный учебник для учащихся старших классов под официальным названием «Информатика и вычислительная техника». Предлагаемая авторами концепция значительно отличается от подхода, принятого в ряде уже опубликованных пособий по той же тематике (например, в учебнике А. Г. Кушниренко, Г. В. Лебедева и Р. А. Свореня «Основы информатики и вычислительной техники», М.: Просвещение, 1993). Вместе с тем, авторы нового издания не могут обсуждать опубликованные учебники и не претендуют на конкуренцию с ними, поскольку предлагаемый материал (несмотря на одинаковое название) посвящен просто иному предмету.

Трудно назвать другую сферу человеческой деятельности, которая развивалась бы столь стремительно и порождала такое разнообразие подходов к обучению, как информатизация и компьютеризация общества. Безнадёжно устарели и фундаментальные монографии, и небольшие книжки, изданные всего 5—10 лет назад и посвященные перспективной, как тогда казалось, информационной технологии на базе мощных ЭВМ с терминалами-дисплеями (как правило, без собственного процессора). Еще 3—4 года назад бесспорным казался лозунг «Программирование — вторая грамотность», под которым подразумевалось умение каждого образованного человека создавать и программировать алгоритмы в своей предметной области на языках ФОРТРАН, Бейсик и т. п. Мощная лавина современных программных продуктов буквально за 1—2 года создала ситуацию, когда в большинстве приложений «кустарное» программирование не только стало ненужным, но и невозможным, — попробуйте, например, «программировать» (в старом смысле слова) в продуктах Microsoft Excel, Access, Word... Со всей очевидностью наиболее актуальным для большинства людей стало не программирование, а умение пользоваться промышленными информационными технологиями (разумеется, речь

не о тех, кто собирается стать профессионалами в этой области, но таких — меньшинство, и учатся они, как правило, в специализированных школах).

С другой стороны, несмотря на динамику технических и программных средств, обозначилась явная тенденция к стабилизации основных технологических подходов к обработке информации: функций клавиатуры, принципов организации и работы с файловой системой, обработки текстов, электронных таблиц, баз данных. Несмотря на все недостатки продуктов фирмы Microsoft (здесь они не обсуждаются), «стихийная» стандартизация пользовательского интерфейса в среде Windows значительно упростила общение с компьютером массы людей, которым ранее приходилось постоянно переучиваться при появлении новых программ.

Вместе с тем значительная часть образованного населения нашей страны по-прежнему испытывает неудобства не столько из-за отсутствия навыков программирования, сколько из-за недостатка знаний об идеологии, методах и возможностях информационной технологии. Результаты этой неосведомленности проявляются в широком диапазоне «компьютерных мифов» — от мистической веры в неограниченные способности машины («надо заложить и посчитать») до наивного недоверия или пренебрежения («откуда машина может знать, что фамилия нашего директора Петров?»).

Поэтому в центре предлагаемого пособия находится информатика как прикладной предмет, цель которого — помочь школьникам освоить основы и возможности современной компьютерной технологии в целом (аппаратура и принципы работы с ней), познакомиться со сложившимся кругом наиболее распространенных операций, выполняемых на ЭВМ. В большинстве предметов, которые учащийся изучает в школе (математика, физика, химия, биология, литература, история, языки и др.), излагаются основы фундаментальных наук, а предметом науки всегда являются объективные, не

зависящие от воли людей, законы природы, общества или художественного творчества. В отличие от таких предметов, большая часть материала, излагаемая в данном курсе информатики, наукой не является, а относится скорее к технологии, которая, хотя и основана на объективных законах, в значительной мере зависит от изобретательности людей, а иногда и просто от их произвола (что особенно характерно для прикладных программ).

Пожалуй, впервые в названии главы школьного учебника появилась аббревиатура фирмы — IBM. Возможно, некоторым такой заголовок покажется недопустимым, однако следует учесть, что эта аббревиатура давно уже стала не столько обозначением фирмы (тем более что вовсе не IBM изобрела персональные компьютеры), сколько знаком, символом современной информационной технологии (хотим мы того или нет).

Три главы учебника посвящены наиболее распространенной триаде операций: обработке текстов, работе с электронными таблицами и базами данных. Авторы старались развеять бытующее заблуждение, что такая информатика (или, если угодно, «компьютерная грамотность») нужна лишь тем, кто в будущем собирается заниматься точными науками или, скажем, бухгалтерией. В частности, умение работать с текстами на компьютере крайне необходимо и школьникам с другими наклонностями — тем, кто собирается стать, скажем, врачом, юристом, историком, журналистом, писателем, композитором или артистом... Впервые в доступной форме рассказано об организации и представлении информации в базах данных (БД), о том, какие возможности открывают БД перед представителями буквально всех профессий (причем не столько научно-технических, сколько «гуманитарных»), о том, как работать с БД, и т. д.

Сказанное выше не означает, что ав-

торы учебника недооценивают необходимость развития алгоритмического мышления школьника. Во-первых, при изложении технологии на многочисленных примерах и задачах разъясняются принципы работы компьютера как автомата, умеющего только феноменально быстро выполнять четыре действия арифметики и сравнивать числа по значению. Пример: если задана дата (число, месяц, год), какие операции должен выполнить автомат, чтобы определить «завтрашнюю» дату? Чем принципиально отличается «мышление» автомата от мышления человека? И т. п. Во-вторых, в специальной главе четко и ясно изложены основы программирования на алгоритмическом языке Бейсик, диалекты которого широко применяются всюду (даже в суперсовременной технологии клиент/сервер).

В отдельной главе школьники знакомятся с основными сферами применения компьютеров в современном обществе (деловые приложения, автоматизация производства и научной деятельности, сети, мультимедиа и др.).

В учебнике сравнительно мало места уделено графической оболочке Windows. Во-первых, значительное количество школ еще не располагает техникой, на которой можно установить Windows; во-вторых, опыт показывает, что пользователи, обладающие базовыми навыками работы с компьютером, осваивают Windows за несколько часов: эта среда отличается удобным и стандартизованным интерфейсом и ориентирована на максимальное упрощение работы.

Разделы учебника сопровождаются вопросами и заданиями; предусмотрено проведение практических занятий с использованием современных прикладных пакетов.

При изложении материала авторы старались предусмотреть максимальную «живучесть» учебника в условиях стремительного технического прогресса.

\* \* \*

## ОТВЕТЫ НА КРОССВОРД, ОПУБЛИКОВАННЫЙ НА СТР. 23

*По горизонтали:* 1. Дисплей. 2. Клавиатура. 3. Цикл. 4. Транзистор. 5. Компьютер. 6. Аргумент. 7. Алгоритм. *По вертикали:* 1. Принтер.



## ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС MultiVision Pro

(USD)

- на базе проекционного оборудования\*
- на базе видео оборудования\*

19.040  
14.400



\* Цены указаны для стартовых моделей мультимедиа комплексов

## MultiMedia КОМПЬЮТЕР

(USD)

- IBM PC/AT-486 - "MultiVision Pro"
- IBM PC Pentium-90 - "MultiVision Pro Master"

3.200  
5.700

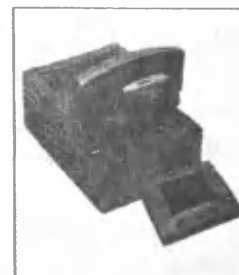


## ПРОЕКЦИОННЫЕ ЖИДКОКРИСТАЛЛИЧЕСКИЕ ПАНЕЛИ

(USD)

- Цветная - Proxima Ovation-840
- Цветная - Proxima Ovation-842
- Цветная - Proxima Ovation-846
- Цветная - Proxima Ovation-920
- DeskTop Projector 2700
- DeskTop Projector 2800

7.399  
8.299  
9.599  
18.199  
10.959  
11.999



## ВИДЕОКАРТЫ

(USD)

- Комплекс видеобработки MIRO

900/1300



## ИНТЕРАКТИВНАЯ СИСТЕМА УПРАВЛЕНИЯ

(USD)

- Радиомышь - RemotePoint
- Сенсорная сканирующая камера - Cyclops 2050
- Сенсорная сканирующая камера - Cyclops 2060

300  
1.199  
1.199

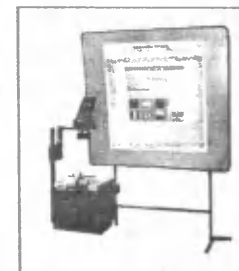


## ПРОЕКЦИОННОЕ ОБОРУДОВАНИЕ И СОПУТСТВУЮЩАЯ МЕБЕЛЬ

(USD)

- Кодоскоп - Overhead Projector Medium 5000
- Кодоскоп - Overhead Projector Medium 10K
- Экран - Medium Data Professional Screen 150x150
- Экран - Medium Data Professional Screen 180x180
- Экран - SuperLight Screen\* 115x150, Stand for Screen
- Экран - SuperLight Screen\* 130x180, Stand for Screen

1.499  
2.399  
370  
499  
1.360  
1.680



\* Рекомендуется использование радиомыши RemotePoint или лазерной указки Laser Pointer в связи с негарантируемой устойчивостью работы указки Cyclops

**Внимание !!!**  
По вопросам приобретения  
и за дополнительной  
информацией обращайтесь:

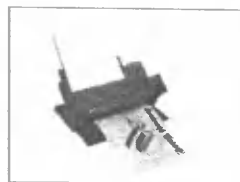
Научный Центр  
Программных Средств Обучения  
Адрес: 103220, г. Москва, ул. Н. Масловка, 16



## ПЕЧАТАЮЩИЕ УСТРОЙСТВА\*

(USD)

- Hewlett Packard Colour Printer - DeskJet 320, 560, 1200C
- Genicom 7025



## СКАНИРУЮЩИЕ УСТРОЙСТВА\*

(USD)

- Hewlett Packard Colour Scanner - ScanJet IIC, IICx
- Envision 8800S

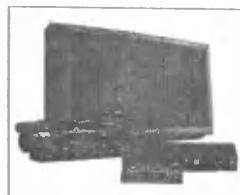


## ВИДЕО ОБОРУДОВАНИЕ\*

(USD)

- Многосистемные видеомагнитофон, видеокамера Panasonic - VHS, S-VHS
- Многосистемный телевизор Panasonic - 14", 29", 33"

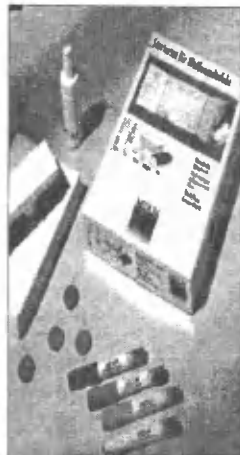
\* Поставляется только в составе комплекса MultiVision Pro



## ДОПОЛНИТЕЛЬНЫЕ АКСЕССУАРЫ

(USD)

- Лазерная указка - Laser Delta 299
- Лампа для проектора Medium 10K 299
- Лампа для проектора Medium 5000 25
- Сумка для переноса Ovation и Cyclops жесткая 404
- Сумка для переноса Ovation и Cyclops мягкая 256
- Сумка для переноса DP и Cyclops жесткая 624
- Сумка для переноса DP и Cyclops мягкая 319
- Набор для чистки проектора 28
- Классная доска Whiteboard standart 120x180 378
- Классная доска Magnetic Whiteboard Exclusive 120x180 548
- Классная доска Cabinet Whiteboard 120x75/120x150 1.119
- Принадлежности для классной доски Startekit 80



## ПРОГРАММНЫЕ СРЕДСТВА

(USD)

- AIST MEDIAMANIA PRO V 1.XX 499
- AIST MEDIAMANIA COMPACT V 1.XX 299
- AIST MEDIAMANIA STUDIO V 1.XX 199
- AIST MEDIAMANIA LITE V 1.XX 99
- CAIRC MULTIVISION V 4.5 249
- CAIRC JUNIOR-1 V 1.0 57
- CAIRC FIVESTARS V 1.0 61
- CAIRC ЗНАНИЯ V 1.1 39
- CAIRC ОРГАНИЗМ ЧЕЛОВЕКА V 1.1 59



### Внимание !!!

Обратившись к нам Вы можете заказать и приобрести практически ЛЮБЫЕ средства для демонстраций, а также для оборудования аудиторий:



тел. (095)214 7784  
факс (095) 214 5433



# APPLE ДЛЯ ОБРАЗОВАНИЯ

А. Г. Блинов,

отдел разработки решений Apple Computer С. I. S.

## ПРОГРАММЫ ДЛЯ MACINTOSH

В этой статье нам хотелось бы познакомить читателей с некоторыми программными продуктами для компьютеров Macintosh. В предыдущих номерах журнала у вас уже была возможность ознакомиться с образовательными программами на русском языке, которые уже созданы (или создаются) в рамках Программы совместных действий Министерства образования Российской Федерации и компании Apple Computer С. I. S. «Технологии Apple в образовании» (см. № 2—3, 1995). Однако сегодня речь пойдет о программах, которые по существу, являются общепользовательскими пакетами или системами разработки, в том числе средствами разработки мультимедиа.

На Macintosh существует изумительный интегрированный пакет, сочетающий в себе одновременно текстовый процессор, электронные таблицы, систему организации баз данных, векторный и растровый графический редактор изображений, а также средство телекоммуникации. Это Claris Works 2.1. Навер-

ное, любой, кто хоть раз работал с этим пакетом, по достоинству оценил его простоту и легкость в обращении. Claris Works 2.1 позволяет использовать все достоинства самого Macintosh.

Claris Works 2.1 на сегодняшний день, пожалуй, самый распространенный интегрированный пакет на Macintosh. Как все интегрированные пакеты в своем классе, он может создавать документы шести различных типов. Каждый тип документа предназначен для определенного вида работы, такого, например, как написание текста или рисование картинки. При создании документа тот появляется в своем собственном окне вместе с нужными для этого типа документа инструментами. Например, над окном текстового редактора находится меню редактирования текста, а под заголовком документа — текстовая линейка. Текстовая линейка нужна, чтобы устанавливать отступы, табуляции, пробелы между строками, выравнивание текста и т. д.

Несмотря на компактность, тексто-



строк) чтобы хранить поступающие данные, которые не помещаются в документе связи. Отметим, что в основном меню Claris Works 2.1 есть с объектом и Сервис. Когда вы меняете оболочки прикладной программы, мему вас был доступ к соответствующим косянкам. Для Российских пользователей фактором является то, что Claris Works 2.1 переведен на русский язык и учрежденный поставляется бесплатно. Для работы с Claris Works 2.1 под компьютер Макинтош, начиная с Macintosh Plus с двумя мегабайтами опе

вая оболочка способна выполнять некоторые операции «больших» текстовых редакторов, например создание собственных стилей текста, или организовывать обтекание текстом объектов и иллюстраций. Оболочка графики оперирует такими объектами, как прямоугольники, круги, линии, многоугольники, и поэтому, содержит богатый набор инструментов для их рисования. Линии сетки вместе с графическими линейками позволяют точно рисовать и выравнивать объекты. Небольшой растровый редактор имеет набор инструментов, необходимых для создания иллюстраций или разработки логотипов. Пакет трансляторов, прилагаемых к Claris Works 2.1, позволяет читать форматы PICT, TIFF, EPS, Mac Paint.

Электронные таблицы Claris Works 2.1 хотя и не обладают таким широким набором средств, как, скажем, Microsoft Excel, но позволяют выполнять ряд распространенных функций, таких, как построение диаграмм для ряда чисел, перенос слов в ячейке, фиксирование строки или столбца, так, чтобы он появлялся на каждой странице в качестве заголовка.

Оболочка базы данных Claris Works 2.1 чем-то напоминает FileMakerPro, только в несколько упрощенном варианте. Инструменты базы данных помогают автоматизировать ввод высших данных в поля базы данных, создавая раскрывающиеся меню выбора, позволяя Claris Works 2.1 вводить для вас величину и проверять данные после ввода. В оболочке базы данных имеется транслятор с языка dBase.

И наконец, оболочка связи представляет достаточно широкий набор удобных и необходимых возможностей: проведение сразу нескольких сеансов связи (каждый сеанс в своем собственном окне), работа с областью переполнения (панель сохраненных строк), чтобы хранить поступающие данные, которые не помещаются в рабочей области вашего документа связи. Отметим, что в основном меню Claris Works 2.1 есть специфические опции — Объект и Сервис. Когда вы меняете оболочку прикладной программы, меню изменяется, чтобы у вас был доступ к соответствующим командам.

Для российских пользователей важным фактором является то, что Claris Works 2.1 переведен на русский язык и для образовательных учреждений предоставляется бесплатно. Российскими специалистами подготовлено методическое

пособие по использованию этого пакета в учебном процессе.

Для работы с Claris Works 2.1 подойдет любой компьютер Macintosh, начиная с Macintosh Plus с двумя мегабайтами оперативной памяти.

В наше время ученики старших классов — это зачастую вполне самостоятельные пользователи компьютеров, которые вместе с преподавателями участвуют в создании собственных программных продуктов или учебных курсов. И кажется вполне разумным, что существование простых и мощных инструментальных средств для создания подобных программ — вещь совершенно необходимая. К счастью, компания Apple предлагает не только ортодоксальные языки программирования. Apple, выпуская свои первые персональные компьютеры, рассчитывала в первую очередь на непрофессионалов в области компьютерной техники и позаботилась о том, чтобы дать им возможность использовать все прелести компьютерных технологий, не углубляясь в изучение языков программирования. Пользователи Macintosh все реже создают традиционные программы на языках высокого уровня — это дело профессионалов. Подавляющее большинство использует уникальные авторские программные средства семейства Card: HyperCard, SuperCard, HyperStudio.

HyperCard фирмы Apple появился с первым Macintosh и стал для этого компьютера тем, чем Бейсик для IBM PC — незаменимым помощником при создании собственных приложений. Освоить HyperCard совсем несложно: с ним работают даже учащиеся младших классов. В то же время вы получаете инструменты для создания приложений, насыщенных текстом, иллюстрациями, звуками, анимацией, фрагментами видеозаписей. Одним словом, в вашем распоряжении оказываются средства создания информационных систем мультимедиа.

Если у вас существуют собственные учебные курсы на IBM PC, вам не составит труда перевести их на Macintosh, не прибегая к программированию в обычном смысле этого слова. Смысл создания собственных приложений заключается в том, что вы создаете проект в виде набора карточек, уложенных в стопку (по-английски — «стек»). Создавая карточки, вы располагаете на каждой из них контрольные кнопки, текст, графику, звуко-

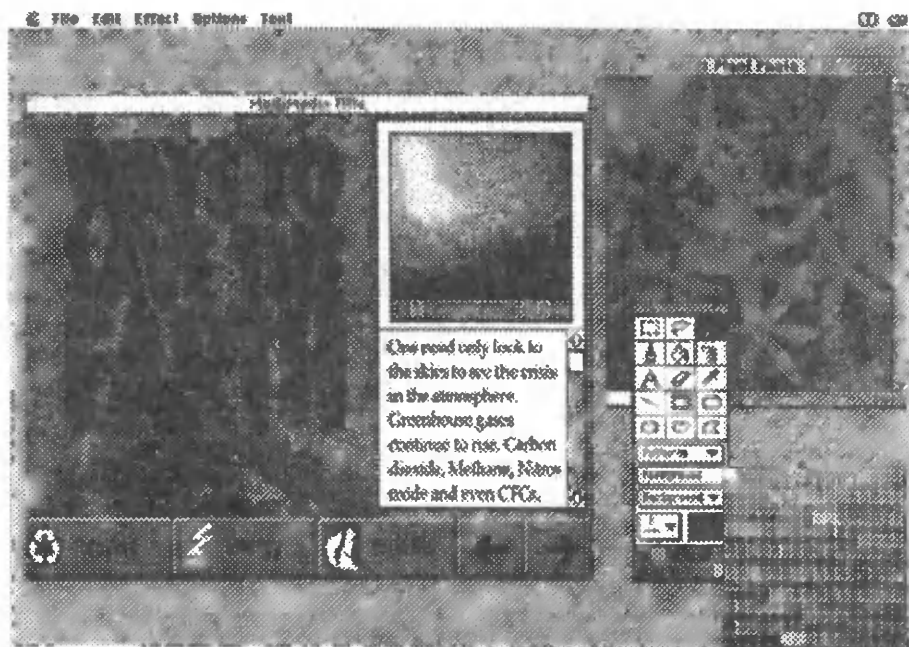
вое сопровождение, анимацию. Стек может занимать весь экран или его часть. После этого вы настраиваете переходы между стеками, и ваш проект готов. На протяжении почти всего процесса создания собственного приложения вам почти не придется обращаться к языку программирования HyperTalk, на котором пишутся все скрипты Macintosh. Да и сам HyperTalk очень напоминает английский язык и прост в освоении.

Работая с HyperCard, вы имеете возможность как пользоваться готовыми стеками, созданными другими авторами, так и проектировать собственные курсы. Каждому стеку может быть присвоен различный уровень прав пользования, для того чтобы избежать изменения вашего проекта другими пользователями. Совсем недавно вышла версия HyperCard 2.3, значительно обогатившая возможности HyperCard 2.2. Теперь возможна работа с палитрой из тысяч цветов, присвоение каждому стеку собственного размера и т. д. Российскими специалистами подготовлено учебное пособие по использованию HyperCard.

Авторские среды HyperStudio 2.0 и SuperCard 1.7 оперируют теми же понятиями, что и HyperCard. Проекты состоят из карточек и называются стеками. Но

есть и отличия: эти среды предъявляют достаточно невысокие требования к свободной оперативной памяти — 1250Кб вместо 4000Кб, которые требует HyperCard при работе с цветом. Усовершенствования касаются в том числе и графических особенностей. Например, возможность включения в стек графических и текстовых файлов без копирования их содержимого. Это незаменимо при просмотре больших по объему картинок и текстов на разных карточках. При наличии соответствующей аппаратуры возможно проигрывание звукового компакт-диска и воспроизведение видеосигнала. Произвольная форма кнопок дает, в частности, возможность реализации интерактивных географических карт. Кнопки могут активизироваться при перемещении в их пределы графического объекта.

Работая с программами семейства Card, вы и ваши ученики наверняка решите если не все, то большую часть ваших насущных проблем. HyperCard 2.3, HyperStudio 2.0 и SuperCard 1.7 прекрасно справляются с поставленной задачей, если вам необходимо создать собственный учебный курс, программу с использованием средств



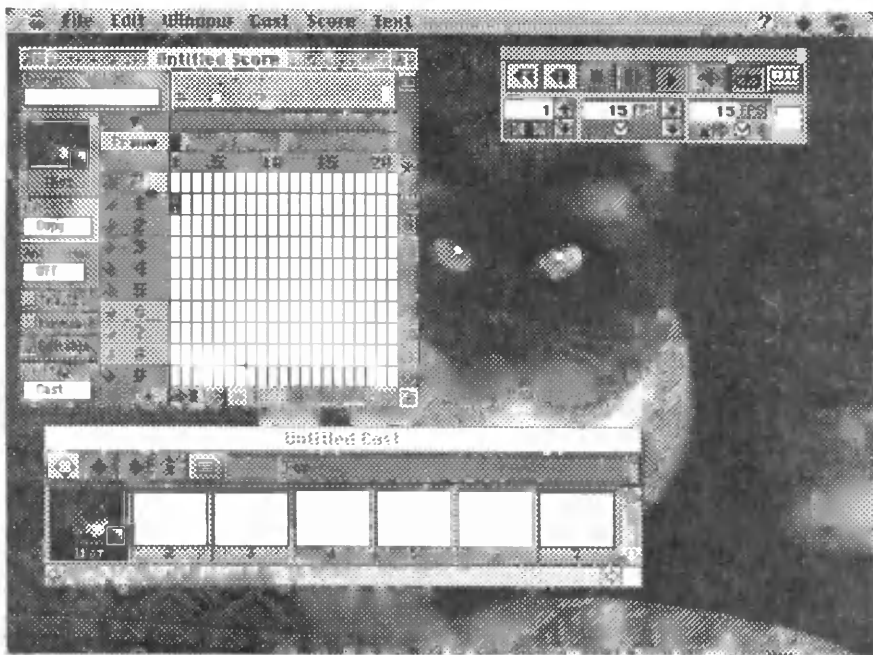
HyperCard 2.3

мультимедиа, использовать эти среды в учебном процессе.

Если вам не хватает интерактивности, движения в ваших программах — на Macintosh существует несколько авторских средств, которые позволяют это сделать. К ним относятся продукты, представляемые фирмой Macromedia (Direcoter 4.0 и AuthorWare Professional 2.2) и Apple (Apple MediaToolkit).

Две программы фирмы Macromedia отличаются по способу их применения. Direcoter 4.0 создавался как авторское программное средство для использования в домашних условиях. Если создатели семейства Card представляли себе вашу программу как набор неподвижных картинок, то программа Direcoter 4.0 подходит к представлению проекта с другой стороны. Для нее то, что получается в итоге, — сценарий, состоящий из набора фильмов. В каждом фильме существуют «актеры», иначе говоря, объекты, которые могут передвигаться, меняться, взаимодействовать друг с другом. В этой связи можно привести хорошее сравнение. Семейство Card является логическим продолжением программ для создания собственных слайд-шоу, а Direcoter — усовершенствованной програм-

мой-аниматором. В программе Direcoter 4.0 есть 11 различных плавающих окон, но для создания собственной простой программы понадобится не больше пяти. Это окно для рисования, окно для «актеров», окно сценария (в нем планируется весь фильм), пульт управления и сцена (на ней «актеры» «исполняют» свои роли). Окно для рисования понадобится для того, чтобы создать или отредактировать собственного «актера». Оно напоминает обычный редактор рисунков типа MacPaint Pro. Когда вы создали собственный рисунок, он автоматически появляется в окне «актеров». Это сделано для того, чтобы вы знали, какими «актерами» вы располагаете. Движение и обработка действий каждого «актера» происходит в окне сценария. Каждый «актер» занимает один или несколько треков или каналов. Пульт управления позволяет просматривать созданный фильм и регулировать различные параметры просмотра, такие, например, как звук. Одновременно с вашими экспериментами с фильмом «актеры» появляются на сцене, и вы можете сразу видеть результаты работы. Direcoter 4.0 требует 4500Кб оперативной памяти и поставляется на CD-ROM. Описание Direcoter 4.0 составлено очень грамотно и продуманно. Просто и четко написанные



Direcoter 4.0

инструкции позволили мне за один час создать собственный интерактивный мультфильм с использованием видеофрагментов. Это авторское средство идеально подходит для создания собственных программ, если вы хорошо знаете, что должно получиться в конечном итоге. В случае, когда вам нужно создавать строго выдержанный и последовательный учебный курс, вам лучше воспользоваться авторским программным средством AuthorWare Professional 2.2.

Программа AuthorWare Professional 2.2 задумывалась как средство для создания именно учебных курсов. Вы создаете некий план, по которому будет идти ваша программа. План чем-то напоминает блок-схему. Создавать ее просто: требуется лишь перетащить и поместить в требуемое место «иконку» с обозначением процесса. Для того чтобы включить в программу такие элементы мультимедиа, как звук или видео, достаточно просто экспортировать необходимый элемент в проект.

AuthorWare Professional 2.2 предлагает богатые возможности для редактирования помещаемых элементов, такие, как проигрывание только части фильма, варьирование его размеров и т. д. Кроме того, AuthorWare Professional 2.2 позволяет помещать часто используемые элементы в библиотеки. Если вы решите изменить элемент в библиотеке, то изменение коснется любой вашей программы, использующей этот элемент. Тот факт, что AuthorWare совершенно спокойно работает как на Macintosh, так и на IBM PC и требует всего 2300Кб памяти, привлекает к этой программе пользователей не только Macintosh, но и других персональных компьютеров.

Есть у AuthorWare Professional 2.2 и несколько недостатков. Один из них — большой размер исполняемых модулей. Разработка той же самой программы на языке типа С или С++ займет примерно в 1,5—2 раза меньше места на диске. Вторым и более существенным недостатком AuthorWare Professional 2.2 является высокая цена: около четырех тысяч долларов. Если вам не по карману такие расходы — воспользуйтесь Director 4.0 или авторским программным средством, разработанным фирмой Apple, — Apple Media Toolkit.

Эта программа, как и AuthorWare, предназначена для написания мультимедийных приложений и очень на нее похожа. Но зато цена у Apple Media Toolkit

в несколько раз меньше, чем у AuthorWare. Правда, Apple Media Toolkit страдает недостатком: программы, написанные с помощью этого авторского средства, не отличаются высокой скоростью.

Все описанные выше программы могут быть использованы непрофессионалами. А как же быть профессионалам? Бытует такое мнение, что программисты при переходе с платформы IBM PC на платформу Macintosh испытывают определенный недостаток в ортодоксальных языках программирования. Этот же вопрос встает и перед учебными заведениями: существуют ли на Macintosh родные сердцу С, С++, Паскаль, Бейсик? На этот вопрос можно смело ответить, что компьютерная грамотность не только не страдает от перехода с платформы IBM PC на Macintosh, а, наоборот, только приобретает новые возможности. Изучение языков программирования и сам процесс программирования на Macintosh выглядит, с одной стороны, значительно проще, чем на PC, а с другой стороны, происходит наглядное обучение выработке единого стиля программирования. Если в среде DOS у вас должна болеть голова об интерфейсе вашей программы, а в среде Windows о всевозможных настройках и установках, то в Macintosh все значительно проще: вам не нужно заботиться ни о том, ни о другом.

Для Macintosh существуют версии С, С++ и Think Pascal от фирмы Symantec и С, С++ от фирмы Zortec. Все компиляторы ориентированы на работу с проектом. Проект — это нетекстовый файл, включающий в себя ссылки на файлы с исходными текстами, файл с ресурсами и библиотеки. Он же содержит скомпилированный код отдельных модулей. Никаких объектных и выполняемых (типа .exe) файлов нет: при пробном запуске программы она загружается в память. Когда программа отлажена, можно также построить отдельную (standalone) прикладную задачу, но на стадии отладки это совершенно необязательно. Не вдаваясь в дальнейшие подробности устройства компиляторов, хотелось бы отметить, что вышеперечисленные программы — это профессиональные системы, полностью соответствующие стандарту ANSI (чем они выгодно отличаются, например, от Borland на IBM PC, который не отслеживает множество ошибок с точки зрения стандарта).

В дополнение к этим компиляторам хотелось бы отметить относительно не-

давнее появление быстродействующего компилятора MetroWerks CodeWarrior, который дает пользователю ряд важных преимуществ. Главным преимуществом можно назвать полный диапазон функциональности в рамках единой среды. Пакет вместе с документацией поставляется на одном CD-ROM, состоит из компиляторов и интегрированных сред программирования (IDE) для трех языков: С, С++ и Паскаля. Каждая IDE включает в себя редактор исходных программ и все существующие средства управления процессом. Интегрированные среды программирования CodeWarrior очень напоминают среды Think компании Symantec, в них управление файлами исходных программ использует описан-

ный выше образ проекта. Редакторы сред дают возможность отмечать ключевые слова и комментарии, чтобы отличить их от остального текста исходной программы.

В следующих номерах журнала мы продолжим знакомить читателей с различными программными продуктами для компьютеров Macintosh. Если же вас интересует более подробная информация или вы желаете лично посмотреть на принцип работы тех или иных программ — обращайтесь к нам, мы будем очень рады ответить на все ваши вопросы.

Наш адрес: 103055 Москва, 2-й Выше-славцев пер., г. 17. Тел.: (095) 978-35-62, 978-80-01, факс: (095) 978-13-91.

## Литература

1. Борисенко В. В. Макинтош: первое впечатление//Информатика и образование. 1994. № 6. С. 34—40.

2. Знакомство с HyperCard: Учебное пособие/Под ред. Д. Д. Рубашкина.

СПб.: Санкт-Петербургский образовательный Apple-центр. 1994.

3. Learning Director//Macromedia. 1994.

4. AuthorWare: Working Model for Macintosh//Macromedia. 1993.

# Apple в образовании

*Все дети Америки и  
Европы учатся на  
Макинтошах.  
А Ваши?*



Поставки компьютерных классов на базе компьютеров Макинтош в школы и детские сады.  
Индивидуальный подбор программного обеспечения.  
Консультации преподавателей по методике обучения на Макинтошах.  
Техническая и методическая поддержка классов.  
Значительные скидки для образования.

## МАК ЦЕНТР



АД МакЦентр — Мастер-дистрибьютер Apple Computer, Inc.

Тел.: 956-68-88 (5 линий+факс), 955-27-20, 955-29-35, Факс: 955-27-79



**А. Г. Блинов,**

*отдел разработки решений Apple Computer C. I. S.*

## КОМПЛЕКТ РУССКОЯЗЫЧНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА CD-ROM

В сентябре 1995 г. компанией Apple Computer C. I. S. при содействии Института новых технологий был создан комплект русскоязычных обучающих программ для компьютеров Macintosh на CD-ROM. Такой CD-ROM бесплатно прилагается к компьютерным классам Macintosh различных конфигураций и содержит самые популярные обучающие программные пакеты с лицензией на класс.

Комплект оформлен на компакт-диске и включает программные продукты и их описания и методические указания по использованию в учебном процессе.

В состав комплекта вошли следующие обучающие программы для компьютеров Macintosh: «Живая физика», «Живая геометрия», «Искусственная жизнь», ЛогоМиры, представленные Институтом новых технологий; КуМир, представленный предприятием ИнфоМир; «Эрмитаж», «В мире русского портрета», «История», представленные центром «Мультимедиа в образовании»; «Юниспелл» и «Маэстро», представленные фирмой «Максимум».

Читатели уже успели ознакомиться с большинством из этих программ (см. № 3, 1995). Сегодня нам хотелось бы рассказать об остальных программах, вошедших в состав комплекта.

«История»: сборник задач по истории для учащихся V—VIII классов, в котором каждая задача построена как мини-исследование одного из аспектов истории России, древнего мира или средних веков. Смысл задач состоит в выборе из достаточно большого количества исходной исторической информации, необходимой для правильного ответа на поставленные вопросы. Ведется подсчет очков, ориентирующий на использование информации, а не на прямой перебор вариантов. Программа не просто закрепляет изученный материал, но и является в прямом смысле слова «мультимедийной». Большое количество изобра-

жений исторических и культурных памятников, фотографии деятелей истории и культуры помогают реализовать принцип «история через искусство». Использование анимации делает программу живой и интересной также и для дополнительного образования.

«Юниспелл»: программа проверки русского правописания и расстановки переносов для ряда популярных программ, таких, как Claris Works, Microsoft Word, Excel, Fox Pro, QuarkXPress, Adobe PageMaker. Программа позволяет проверять правописание по словарю, насчитывающему 100 000 словооснов, что составляет около 3 млн. слов.

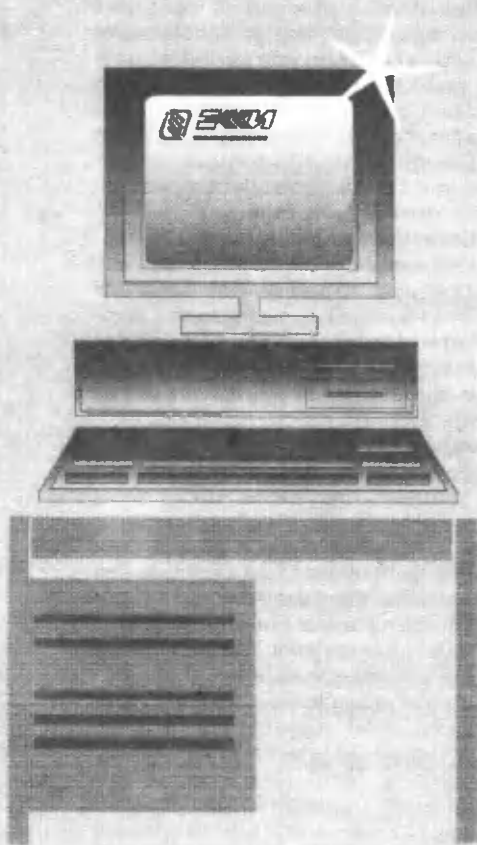
«Маэстро»: очень симпатичная программа обучения машинописи, больше напоминающая игрушку. За короткие сроки «Маэстро» позволяет достичь весьма впечатляющих результатов при печатании текста слепым способом.

Большинство из перечисленных выше программ были одобрены Министерством образования Российской Федерации в результате экспертизы, проводимой в рамках Программы совместных действий Министерства образования и компании Apple Computer C. I. S. Эти программные пакеты были официально рекомендованы к использованию в учебном процессе письмом Министерства образования РФ от 3 июля 1995 г. в региональные органы управления образованием, институты повышения квалификации работников образования, а также педагогические университеты и вузы России.

В комплект русскоязычных обучающих программ для компьютеров Macintosh на CD-ROM мы также постарались включить некоторое количество демоверсий, наиболее популярных в мире образовательных программ. Более подробную информацию о данном комплекте вы сможете получить, связавшись с нами по телефону: **978-35-62** или **978-80-01**.

*Вы знаете,  
какой должна быть  
Ваша школа завтра.  
Мы сделаем ее такой уже сегодня.*

**ЭКСИ - комплекс услуг по информационному и техническому обеспечению учебных заведений**



**К**лассы информатики, физики, химии, домоводства, лингафонные кабинеты и т.д. "под ключ"

**О**хранная сигнализация, установка металлических дверей и решеток

**М**ебель стандартная и по индивидуальному заказу

**П**рограммный продукт - разработка и распространение

**Л**окальные сети, мультимедиа-классы

**Е**жемесячное сервисное обслуживание ВТ, оргтехники

**К**омпьютеризация административных каналов управления

**С**истематизированный банк прикладных программ для ИВМ, УКНЦ, БК

# ПРОФЕССИОНАЛЬНЫЙ КОМПЬЮТЕР

Т. С. Терновская,

*Петрозаводский государственный университет*

## ВОПРОСЫ МАРКЕТИНГА ПРОГРАММНЫХ ПРОДУКТОВ

Региональный центр новых информационных технологий при Петрозаводском университете, специализирующийся на разработке программно-методических комплексов автоматизированных систем обучения и управления для организаций, предприятий и учебных заведений, активно занимается вопросами маркетинга программных продуктов (ПП), которые оказались очень актуальными в условиях формирования рынка.

При рассмотрении вопросов маркетинга программных продуктов большое внимание должно уделяться проблемам предотвращения «программного пиратства», соблюдению законодательства, лицензионных соглашений, определению стратегии продаж и тиражирования программ, а также сервисного обслуживания разработанных комплексов.

На рынке любой программный продукт имеет свой жизненный цикл или период рыночной устойчивости, который состоит из следующих стадий развития: внедрение, рост, зрелость, насыщение, спад.

Период внедрения на рынок отличается тем, что продажа программного продукта на этой стадии убыточна, так как объем продаж низок, а маркетинговые расходы велики (особенно на рекламу). Здесь разработчик должен уделять внимание качеству программного продукта, вести его доработку и совершенствование. Если вы приобретаете программный продукт, который находится на этой стадии, у вас могут возникнуть проблемы с его эксплуатацией: он может неожиданно «падать» или «зависать». Таким образом, вы будете вынуждены снова и снова обращаться к разработчикам. Поэтому в договоре на покупку программного продукта обязательно должен быть записан пункт о бесплатном сопровожде-

нии ПП в течение года или другого срока.

Если программный продукт вышел на стадию жизненного цикла, которая называется «рост», то это означает, что он успешно работает на рабочих местах и на него увеличивается спрос. При этом объем продаж растет, прибыль увеличивается, расходы на рекламу стабилизируются. Разработчик при этом, как правило, уделяет большое внимание совершенствованию дизайна.

Далеко не каждый программный продукт выходит на стадию зрелости. Если это происходит, то он признан и широко используется. В этом случае разработчик получает максимальную прибыль и может снижать цену, а покупатель доволен качеством разработки и без труда внедряет программу в своей организации.

Когда программный продукт выходит на стадию насыщения и рост продаж прекращается, разработчик может получать прибыль за счет уменьшения собственных расходов, связанных с его тиражированием.

На последней стадии развития жизненного цикла программного продукта, которая называется спадом, разработчик, чтобы найти других пользователей и поддерживать спрос, должен перейти на новый рынок или внести существенные корректировки в программу.

Нахождение программного продукта на рынке обусловлено до тех пор, пока издержки не превышают прибыль. При этом конкуренты могут внести существенные коррективы в жизненный цикл, так как новый программный продукт конкурентов способен замедлить рост продаж вашего товара, остановить выход на стадию насыщения и сразу перевести в стадию спада. Поэтому показатели конкурентоспособности программы необхо-

димо задавать с большим запасом, с учетом будущей конкуренции, тем более что срок рыночной устойчивости программного продукта невелик по сравнению с другими товарами.

На зарубежном рынке срок жизни хорошего программного продукта находится в пределах от 6 месяцев до 1 года. По истечении этого срока он или корректируется и появляется его новая версия, или уходит с рынка программных продуктов и больше не продается.

Чтобы выяснить, на какой стадии развития жизненного цикла находится программный продукт, рекламному агенту нужно выяснить, сколько продаж этой программы осуществила фирма, узнать организации и телефоны тех пользователей, которые уже купили его, и выслушать их мнение о данном программном продукте.

Если разработчик не может предоставить вам эту информацию, значит, вы — один из первых клиентов, и эта программа находится на начальной стадии своего жизненного цикла. Таким образом, в случае покупки у вас могут возникнуть проблемы с ее внедрением.

Солідные фирмы — разработчики программных продуктов первые варианты новых программ передают пользователям бесплатно или продают по символической цене, чтобы покупатели согласились на совместное сотрудничество с ними по устранению ошибок или по дальнейшему развитию и дополнению комплекса программ.

При продаже программных продуктов используются, как правило, четыре стратегии маркетинга: интенсивный маркетинг, выборочное проникновение, широкое проникновение, пассивный маркетинг. При определении стратегии принимаются в расчет не только цена продукта, расходы на стимулирование сбыта, качество программного продукта, но и конкуренты. Все это вносит коррективы в определение оптимальной стратегии.

За рубежом производством программных продуктов занимаются коммерческие предприятия, специализирующиеся в области создания программных средств:

- небольшой разработчик прикладных систем по индивидуальным заказам;

- большой разработчик заказных прикладных систем в национальном масштабе;
- предприятие, разрабатывающее и широко распространяющее программные пакеты;
- предприятие, разрабатывающее проблемно-ориентированный настраиваемый пакет с узкой специализацией;
- предприятие, разрабатывающее объекты-компоненты программных средств;
- предприятие, создающее по заказу клиентов прикладные системы на основе доступных объектов — компонентов и пакетов;
- софтверные сообщества.

Эта классификация применима и для российского рынка производителей программных продуктов.

При производстве и продаже программных продуктов необходимо учитывать наличие лицензий на программные средства и положения закона «О правовой охране программ для ЭВМ и баз данных», который определяет программу для ЭВМ как форму представления совокупности данных и команд, предназначенных для функционирования ЭВМ, а также подготовительные материалы, полученные в ходе ее разработки и порождаемые ею аудиовизуальные отображения.

С расширением круга пользователей региональных и глобальных информационных сетей, использующих технологию открытых систем, центр тяжести работ, связанных с маркетингом программных продуктов, смещается в область международного сотрудничества. Первостепенное значение приобретает налаживание работы по «сетевым» услугам как со стороны отечественных разработчиков, так и со стороны иностранных партнеров. Более актуальной становится работа с новейшими и наиболее эффективными стандартизованными средствами (среда разработчиков), кооперация в разработке больших программных комплексов, баз данных, учебных средств, согласование прав на интеллектуальную собственность и по другим вопросам функционирования в среде «глобального интеллекта».

## ОТВЕТЫ НА КРОССВОРД, ОПУБЛИКОВАННЫЙ НА СТР. 97

По горизонтали: 1. Компьютер. 2. Монитор. 3. Диск. 4. Разрядность. 5. Джойстик. 6. Процессор. 7. Программа. По вертикали: 1. Принтер.

# НОВЫЕ ПРОЕКТЫ

**В.А.Урнов,**

президент НЦПСО—АИСТ group

## «ВДОЛЬ ПО ДОРОГЕ...»\*

*«Можно, конечно, объявить кампанию по предотвращению зимы, шаманить, нажравшись мухомора, бить в бубны, выкрикивать заклинания, но лучше все-таки шить шубы и покупать валенки...»*

А. Стругацкий, Б. Стругацкий

«Улитка на склоне»

Итак, прошло ровно 10 лет с того момента, когда впервые в нашей группе у двоих выпускников мех-мата МГУ (Д.Ю.Климов — научный сотрудник закрытого «ящика» и я — учитель школы N 209 Свердловского района) появилась идея создания «учительского компилятора» — программы, специально созданной для написания других программ. Первым «прототипом» программной реализации этой идеи была написанная на Бейсике ДВК-2 программа, опубликованная в дальнейшем в нашем учебнике «Преподавание информатики в компьютерном классе» (М.: Просвещение, 1990). Этот простейший транслятор требовал от учителя исходных данных в виде вопросов и вариантов ответов на них с указанием правильного номера ответа. На выходе получалась программа-тренажер для ученика. Понятно, что и вопросы и ответы каждый раз «перемешивались», дабы избежать возможности «тупого» запоминания. Результатом работы тренажера была пара данных — фамилия и оценка. Например: «Д.Ю.Климов — 5» или «В.А.Урнов — 2» и т.д. В связи с весьма и весьма ограниченными возможностями компьютера ДВК, в программе использовалась исключительно «текстовая» информация, графика отсутствовала (рис.1). Это был 1986 год. Своей работой мы, по молодости, остались удовлетворены и решили двигаться дальше.

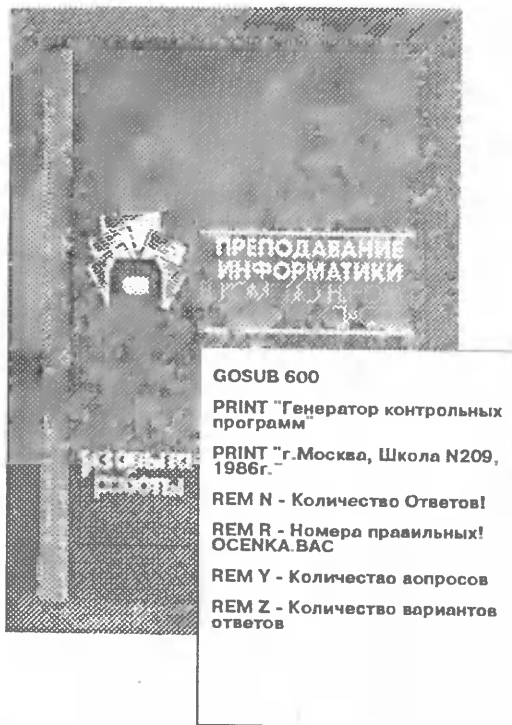


Рис.1. Фрагмент учебника

\* Из кинофильма «Старая, старая сказка...»

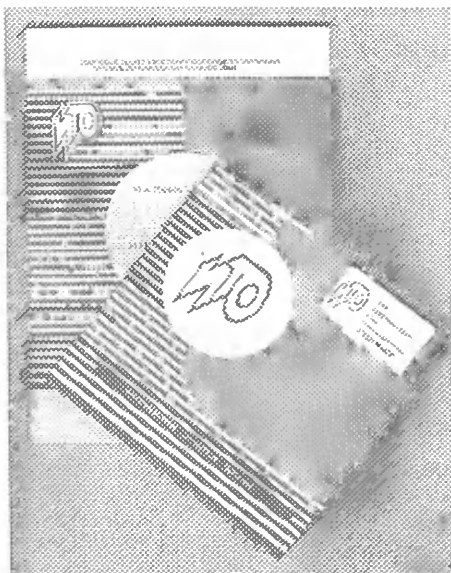


Рис. 2. LOGO

Norton Commander или какую-другую DOS-оболочку, а также пару игрушек (в CGA режиме хуже, чем на «Ямахе») — и все! Уверен, что с 1989 по 1992 годы реальной пользы от классов УКНЦ в школах было гораздо больше, чем от IBM! Причина проста: наличие программных средств.

Если коротко перечислить только программы для УКНЦ, создание которых финансировалось и координировалось НЦПСО в этот период, получится весьма внушительный список:

- *тренажер «MIT LOGO» (Климов, Петров, Чистопольский);*
- *текстовый редактор «Writer» (Золотуский, Миронов, Трикузов);*
- *сетевой монитор «ВАМ» (Махуха);*
- *УКНЦ — версия «Роботландии» (Первин, Гольцман, Зайдельман);*
- *УКНЦ — версия пакета «Информатика — 89» (Меняйло, Щекочихин);*
- *пакет «Игрушка» (Баранов, Золотуский, Миронов) и т.д.*

Кстати, некоторые из наших пакетов, такие как сетевой монитор, драйвер электронного диска и другие затем «вставлялись» в классы УКНЦ как «стандартный» набор программного обеспечения безо всякой ссылки на НЦПСО.

Хочется привести один смешной пример. В 1991 году мы выставили наши пакеты на конкурсе ВДНХ и получили первую премию за LOGO и несколько вторых за другие пакеты (для УКНЦ и «Ямахи»). Наш ответ об авторстве программ — компания НЦПСО — почему-то не удовлетворил руководство ВДНХ. Нас попросили указать одну(!) персону конкретно как автора пакета. Крайне удивленные таким подходом, не восприняв его всерьез, мы для первой премии за транслятор «MIT LOGO» «создали» из четырех авторов — Д.Ю.Климова, М.К.Петрова, В.А.Урнова и П.А.Чистопольского — одного. Имя ему было дано, соответственно: Клим Петрович Урнопольский. Каково же было, представьте, наше удивление, когда месяца через 3-4 мы получили золотую медаль ВДНХ, диплом на имя К.П.Урнопольского, а также рекомендацию руководству НЦПСО выплатить ему премию в размере

Техническим решением для московских школ в период с 1988 по 1990 годы оказался компьютер УКНЦ. Графика на этой микро-ЭВМ была, но черно-белая (монитор!), хотя и быстрая. Нашим следующим шагом была реализация для УКНЦ компилятора LOGO Пейперта. Видимо, также удачная, поскольку и сам компилятор, и различного рода документация к нему, блестяще написанная в своей методической части Ю.А. Первиным, используются и по сей день. По сути, это была первая программная разработка, сделанная в рамках НЦПСО нашим коллективом (рис.2).

1990—1992 годы. «IBM — в обзавание». Классы PS/2, 286 машины — валом на рынке. Программных средств, как водится, никаких! EGA-мониторы, 16 цветов, надежно, красиво, «импортно». Можно запустить

300 руб. Премию мы платить не стали, а вот диплом сохранили до сих пор (рис.3).

Однако все созданное нами в период с 1987 по 1991 годы удачно подходило для курса информатики, но никак не годилось для использования в предметных областях. Продолжая думать о возможной реализации «не-программистского» транслятора уже для IBM, в 1991 году на основе форматов графического редактора Dr. Genius в НЦПСО был создан пакет «исполняемые» (\*.exe) файлы, пользуясь исключительно кнопками «мыши». В 1992 году, продолжая идеи, реализованные в «Designer», совместно с компанией IST AB (Швеция) был создан пакет «TeachCAD». Были произведены две законченные версии этого транслятора — TeachCAD v.3.0 и v.3.5. Пакет поставлялся, в основном, в московские школы. Он содержал в себе встроенные редакторы — графический и текстовый, «библиотекарь» для работы с готовыми наборами «картинок» и cliparts (т.е. сами библиотеки предметно-ориентированных «картинок»). Пакет устойчиво работал на 286 и, позднее, на 386 компьютерах при ОЗУ 640 Кб и был рассчитан на EGA разрешение в 16 цветах.

По нашим сведениям, всего «официально» было поставлено около 500 пакетов. Более чем в 100 школах пакет используется до сих пор. Столь же успешно его до сегодняшнего дня нелегально продают и на рынках (рис.4).

1993 год. Идеи компьютерной анимации и робкие надежды на возможность работать с «живым видео», а также насущные потребности расширения возможностей пакета «TeachCAD» (необходимость перехода на «Windows-интерфейс», возможность компиляции под «Windows») привели к появлению пакета «MultiVision v.4.0» (MV). Основное его отличие от «TeachCAD», кроме в корне измененного интерфейса, — появление «полуавтоматического» аниматора и «библиотекаря» мультфильмов.

MV «Animator» на основе «апликационной» технологии позволяет готовить «мультик» по ключевым кадрам, заданной траектории и указанным для ключевых кадров линейным и нелинейным искажениям. «Библиотекарь» дает возможность менять на экране взаимное расположение отдельных «мультиков», их продолжительность и положение во времени

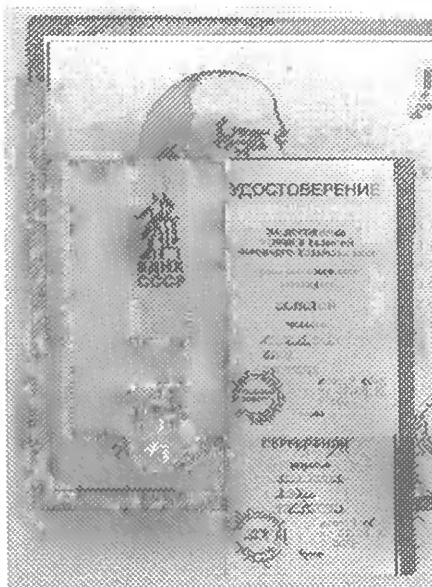


Рис.3. Диплом

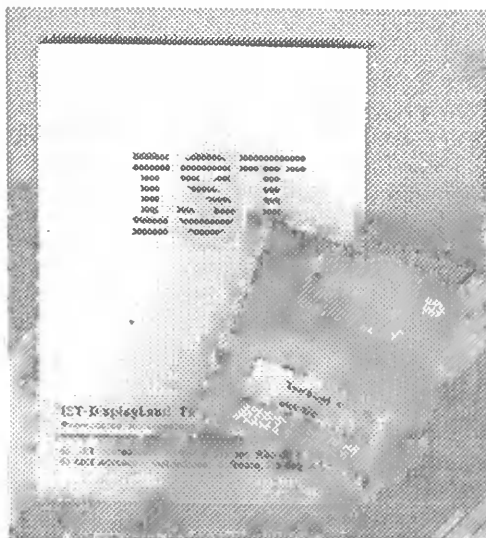


Рис.4. «TeachCAD»



Рис.5. MV

1995 году (рис.6).

Основой всех пакетов для IBM — от «Disigner v.2.0» до MV v.4.5 являлось ядро транслятора, организованное на следующих простейших принципах: на экране отмечается произвольная четырехугольная «зона». Таких зон может быть сколько угодно, единственное ограничение — они не могут пересекаться. У каждой «зоны» есть три «слота», и к каждому по желанию пользователя можно «привязать» одно из дальнейших действий аналогично тому, как к «тройнику» можно подключить одновременно три различных электрических прибора. Фактически, с 1991 по 1995 годы мы просто увеличивали набор «бытовых приборов» — от текстовых окон и картинок до библиотек мультфильмов, звуковых файлов и фрагментов «живого видео». Базис оставался неизменным. Тем самым пользователь, научившись однажды работать с «Disigner» или «TeachCAD», не испытывал никаких трудностей и со всеми дальнейшими версиями этого единого семейства пакетов. Просто в его руках становилось все больше и больше различных мультимедиа-инструментов и подсобного «полуфабрикатного» материала. Это давало ему возможность сделать свою собственную «авторскую» программу все более и более гуманизированной, яркой, доходчивой и профессиональной, т.е. сделать всю необходимую информацию и методические идеи более доступными для восприятия учениками.

Читателю также хорошо известно, что с 1992 года НЦПСО и другие компании CAIRC-AIST group детально прорабатывали вопрос соответствующей аппаратной поддержки подобного учительского АРМа (см., например, ряд статей в ИНФО 1994-1995 годов), названного «MV Pro комплексом». Семейство мультимедиа-комплексов MultiVision Pro, на наш взгляд, дает в руки учителя все необходимое для использования MV-оболочки в любой предметной области. Достигнутая ценовая эффективность комплекса с 25 000-30 000 USD в 1993 году до 14 000 USD в 1995 при полной замене

относительно друг друга, расположение «по слоям» (кто «над» кем, а кто «под») и т.д.

Берусь утверждать, что MV v.4.0 стал первым отечественным мультимедиа-пакетом, созданным специально для образования. Эта версия также производилась совместно с компанией IST AB (рис.5).

Двигаясь дальше в том же направлении, к 1994 году в пакет «MultiVision» были добавлены работа со звуком (v.4.3), а также возможность включать фрагменты «живого видео» (из FLI формата). Следуя многочисленным требованиям пользователей, в пакет была также включена программа генерации тренажеров и опросников (как и все подобные программы мало отличающаяся по своей сути от Бейсик-версии 1986 г.), а также большое количество специально созданных библиотек изображений и видеофрагментов по различным предметам. Была, кроме того, переработана в сторону упрощения и документация пакета. В связи с тем, что объем пакета сильно вырос, в первую очередь, за счет «видео», было принято решение о выпуске MV v.4.5 на CD-ROM, который и находится на рынке сегодня — в



всех комплектующих на «сегодняшние» и даже «завтрашние» делает MV-технология реально доступной, высокопроизводительной по параметрам качество — эффективность — цена для любой школы России.

Чего мы хотим добиться в ближайшем будущем? Что ждет пользователя нашей продукции во второй половине 1995 и в 1996 году? Попробуем дать четкие ответы на эти вопросы.

Вплотную занявшись компьютерной анимацией и видеообработкой, мы волей-неволей оказались и на других рынках программных средств, кроме образовательного. С 1993 по 1995 годы компаниями нашей группы (НЦПСО-Комплекс и АИСТ Инс.) было создано семейство программ «MediaMania» в различных вариантах — от упрощенных («Special Edition» и «Lite») до профессиональной версии для таких рынков, как «домашнее видео» и рынок компьютерной видео-анимации.

«Откатка» этих пакетов на более избалованных вниманием пользователях, а также резкое удешевление и упрощение аппаратных блоков видеообработки на мировом рынке позволила нам спланировать выход в конце 1995 — начале 1996 года нового пакета для образования под условным названием «Проект 5». Этот пакет вновь основывается на базовой идее «зонного» компилятора, но реализуется в среде «Windows 3.11», а также «Windows 95». Новый пакет будет содержать кроме ядра — «Disigner» — пользовательский «Animator», который по своим возможностям вполне соответствует видеомашинам, используемым на телевидении, например, для монтажа новостей и т.д.

Кроме того, в пакет будет включен и модуль автоматической анимации, т.е. система создания мультипликационных фрагментов на базе подготовленных самим пользователем слайд-шоу или использования готовых темплетов (например, из «MS PowerPoint 4.0»). «Проект 5» содержит также и полный набор программ для работы с «живым видео», таких как: «Video Capture», «Composer», «Video Edit», «Morphing», «Video Player», поддерживающих все наиболее распространенные видеоформаты.

Одной из очень важных для нас задач является попытка добиться совместимости между существующими сегодня версиями MV и новой программой. Это позволило бы и нам и нашим постоянным покупателям использовать уже наработанные предметные библиотеки и целые курсы в



Рис.6. CD-ROM MV

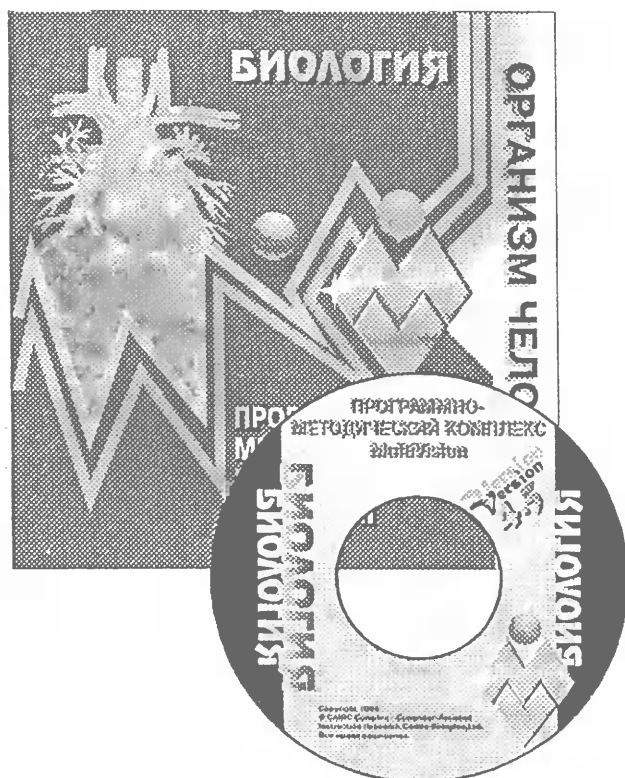


Рис.7. «MV-Биология»

Параллельно с работой над «Проектом 5» мы планируем продолжить производство «предметных» CD-ROM. Предположительно, это будут сначала Экология и География, а далее — Химия и Литература.

Основной отличительной особенностью материалов на подобных «предметных» носителях, является то, что учителю — Автору своего собственного курса предоставляется информация для использования в той области всего необъятного учебного процесса, где он считает нужным, и в том объеме, который он сам сочтет необходимым, а не как это обычно бывает — фрагментарные, жестко заданные выборочные программы-уроки (см. статью Е.Ю. Тюменцевой «MV и MV Pro в учебном процессе». — ИНФО. — 1995. — № 2.).

В заключение хотелось бы отметить, что все перечисленные выше программные пакеты были «отработаны» нашими компаниями от самого начала (основных идей, первых реализаций и версий) до малого промышленного производства и маркетингования. Хочется верить, что выбранное нами направление дает действительное решение по внедрению информационных технологий в учебный процесс. Конечно, единственное абсолютное суждение здесь за нашим пользователем, нашим покупателем, с которым мы находимся рядом вот уже 10 лет и которому мы глубоко признательны за эти годы, пройденные вместе.

рамках нового пакета, расширяя и увеличивая их эффективность за счет новых возможностей. Мы надеемся приступить к поставкам нового пакета не позднее февраля-марта 1996 года.

До конца же 1995 года нами заложены два новых предметных CD-ROM, основанных на MV-технологии. Это программно-методические комплексы «MV-Биология» (рис.7) и «MV-История». В их состав входит большое количество готовых и «полуфабрикатных» материалов по данным предметам, а также программы-tutor(ы) и видеокассеты по использованию MV-технологии как в производстве конкретных курсов, так и непосредственно в учебном процессе.

Д. В. Белич,

преподаватель компьютерных технологий и экономики школы № 216

## СМЕНИМ ДОСКУ И МЕЛ НА ЭКРАН И ЛАЗЕР

После появления в нашей школе комплекса MultiVision Pro открылись совершенно неожиданные возможности в подготовке и преподавании самых различных курсов. Преподавание с помощью комплекса операционной среды Microsoft Windows и основных приложений к ней, таких как Works, Word, Excel и PowerPoint с набором специально разработанных примеров и заданий проходит несравнимо живее и плодотворнее. Ученики имеют возможность постоянно следить за действиями учителя, а не сидят к нему спиной, уткнувшись в экраны своих компьютеров.

Однако помимо того неограниченного вклада, который MultiVision Pro внес в преподавание компьютерных технологий, комплекс позволил поднять на новый качественный уровень подготовку и проведение занятий и по другим предметам.

В нашей школе вторым профилирующим предметом является экономика, поэтому основная часть материалов связана с ней. Скажу больше - подготовлен целостный курс преподавания экономики с применением компьютерного демонстрационного оборудования. В основном материалы связаны с курсом "Прикладная экономика", разработанным корпорацией Junior Achievement. Сложные графики и диаграммы крупным планом в цвете теперь появляются по мановению красного лучика — лазера. Кроме того, в течение нескольких минут они могут быть изменены по форме или содержанию. Образцы бланков и иных документов заполняются прямо на большом экране, при этом ни на секунду не теряется прямой контакт с учениками. Обсуждение графических решений задач происходит без томительного ожидания различных геометрических построений: готовые кадры сменяют друг друга, дополняя чертеж все большими подробностями, приближающими класс к правильному ответу. Чередой предстают метровые доллары, марки, франки, лиры; а учителю достаточно всего лишь попасть лучом в президента, изображенного на купюре, чтобы появилось имя и подроб-

ности его биографии. Так же просто увидеть водяные знаки, защитные полосы и рассмотреть купюру почти изнутри.

Или, скажем, объясняя такую науку как делопроизводство, разве не приятно, щелкнув в уголок листа, увидеть там появившийся герб или штамп, а в другом месте печатать и подпись, а также заголовки и разные части текста. Причем все это вместе с педагогом видит весь класс. И общается учитель не с макушками детей, уткнувшихся в образцы, лежащие на партах — все смотрят только на него, ну и на экран, конечно.

Многие учителя, традиционно обходящие компьютерный кабинет стороной, с появлением комплекса MultiVision Pro внезапно проявили к нему повышенный интерес. Возможность рассмотреть крупным планом в трехмерном изображении внутренние органы человека, путешествие внутри организма, а также набор статических и движущихся изображений динозавров буквально поразили преподавателя биологии. Учитель истории нашел в электронной энциклопедии портреты всех президентов США и узнал о них много такого, о чем раньше не имел понятия. Чего стоят хотя бы кадры хроники встречи Хрущева с Никсоном или подписания акта капитуляции Японии на борту Миссури! «Живая книга», герои которой разговаривают с вами на трех языках по выбору, лишила сна «язычников» (не по вероисповеданию, а по профессии). Ну а самым большим поклонником мультимедиа-техники стал, пожалуй, преподаватель музыки после того, как он вывел на экран изображение всех известных ему инструментов и легким попаданием луча заставил их звучать.

Исключительно важную роль имеет возможность распространения наработок, сделанных учителями школ, обладающих подобной техникой. Вот почему я обращаюсь с призывом к сотрудничеству к тем, кто хочет «успеть в будущее» даже не обладая всеми необходимыми техническими средствами сегодня. Ведь проблема начинает решаться тогда, когда вы осознали необходимость ее решения.

Международная академия информатизации  
Ассоциация учителей и преподавателей информатики  
Московский городской физико-математический лицей № 1511 при МИФИ  
КНПП БИТ

**4-я Международная конференция-выставка  
«Информационные технологии в образовании» — «ИТО-95»  
состоится с 28 ноября по 1 декабря 1995 г.**

В программу конференции будут включены пленарные, секционные, стендовые доклады и дискуссии по следующим направлениям.

**Секция А:** Содержание и проекты стандарта образования по информатике

**Секция В:** Информационные технологии в преподавании

**Секция С:** Информационные технологии в управлении

**Секция D:** Телекоммуникации, дистанционное обучение

**Секция Е:** Мультимедиа технологии в образовании и искусстве

Во время конференции будет проходить организационная конференция Ассоциации учителей и преподавателей информатики.

**Срок представления докладов и заявок на участие в конференции не позднее 21 октября 1995 г.** Справки по оформлению тезисов по телефону (095) 186-26-01.

Для регистрации необходимо выслать в адрес Оргкомитета по обычной или электронной почте заполненную заявку (регистрационную форму) с указанием номера платежного поручения.

**Организационный взнос**, обеспечивающий получение материалов конференции, составляет **20 долларов США**.

Организационный взнос следует перечислить на расчетный счет ООО «КНПП БИТ» с указанием фамилии и инициалов участника и пометкой «За участие в ИТО-95».

Все материалы направлять в адрес Оргкомитета.

*Для расчетов по Москве и Московской области:*

расчетный счет № 467468 в филиале «Экспресс» ММКБ, МФО 997072, уч. 4В.

*Для расчетов с другими городами СНГ:*

расчетный счет № 467468 в филиале «Экспресс» ММКБ, № 706161200 в РКЦ ГУМО ЦБ РФ, МФО 211004, уч. СЗ.

Питание и проживание оплачивается участниками конференции самостоятельно. По желанию участников Оргкомитет может забронировать место в гостинице. Оплату бронирования места необходимо согласовать с Оргкомитетом.

**Место проведения конференции:**

115522, Москва, Пролетарский проспект, дом 6, корпус 3.

**Оргкомитет:**

**Адреса:** 129281, Москва, Старо-Ватутинский проезд, дом 8      115522, Москва, Пролетарский проспект, дом 6, корпус 3, КНПП БИТ

**Телефоны:** (095) 186-26-01      (095) 324-55-86  
для справок по вопросам участия в конференции      для справок по вопросам участия в конференции, аренды машинного времени и участия в выставке

**Электронная почта:** office@combit.msk.su

В рамках конференции будет проводиться коммерческая выставка-продажа программных и технических средств, а также методических разработок. Принимаются заявки на участие в выставке. Оплату аренды машинного времени и участия в выставке необходимо согласовать с Оргкомитетом.

Регистрационную форму и другие материалы можно получить в Оргкомитете.

Научно-методический журнал  
Учрежден Министерством  
образования РФ  
и коллективом редакции

Издается с августа 1986 г.  
Выходит шесть раз в год

Главный редактор  
академик  
**БЕЛОЦЕРКОВСКИЙ О.М.**

Заместитель главного  
редактора  
**КРАВЦОВА А.Ю.**

## РЕДАКЦИОННАЯ КОЛЛЕГИЯ

**Болотов В. А.**  
**Бредихин Г. А.**  
**Васильев Б. М.**  
**Зайдельман Я. Н.**  
**Зубченко А. А.**  
**Киселев Б. Г.**  
**Королев В. А.**  
**Кравцова А. Ю.**  
**Краснов А. Я.**  
**Кузякин А. П.**  
**Курнешова Л. Е.**  
**Лапчик М. П.**  
**Леонов А. Г.**  
**Пахомова Н. Ю.**  
**Савин А. Ю.**  
**Самовольнова Л. Е.**  
**Сапрыкин В. И.**  
**Смекалин Д. О.**  
**Уваров А. Ю.**  
**Угринович Н. Д.**  
**Урнов В. А.**  
**Фурсенко А. И.**  
**Хорошилов В. О.**  
**Христочевский С. А.**  
**Чуриков П. А.**  
**Щенников В. В.**

## СОДЕРЖАНИЕ CONTENTS

### МОСКОВСКИЙ ВЫПУСК MOSCOW ISSUE

|  |    |
|--|----|
| Курнешова Л. Е. От инвестиций в образование зависит будущее России<br>Kurneshova L. Russian Future Depends on Investments to Education   | 1  |
| Семенов А. Л. Образование, информатика, компьютеры<br>Semenov A. Education, Information Sciences and Computers   | 5  |
| Переверзев Л. Б. Полюбить машины, помогающие учиться<br>(Образовательная философия Сеймура Пейперта)<br>Pererverzev L. To Love the Machines which Helps us to Learn<br>(Educational Philosophy by Seymour Papert)  | 12 |
| Сопрунов С. Ф. Лого-среда для детского творчества<br>Soprunov S. Logo-environment for Children's Creativity  | 24 |
| Кутукова О. Г. Методика подведения итогов<br>Kutukova O. A Procedure for Results Counting  | 26 |
| Станченко Н. С. Фигуристы, клумбы и Чучело —<br>образные ассоциации в ЛогоМирах<br>Stanchenko N. Figure Skaters, Flower-Beds and Scarecrow<br>as Evocative Associations in MicroWorlds   | 29 |
| Малышева М. А. Дети и Лого<br>Malysheva M. Children and Logo   | 33 |
| Юдина А. Г. История с зайцем<br>Yudina A. A Story of a Hare  | 36 |
| Горбунов А. Л. Компьютерные сети для детей через Lego TC Logo<br>Gorbunov A. Networks for Children based on Lego TC Logo   | 44 |
| Митина О. В. Программная среда Лого глазами психолога<br>Mitina O. Logo Environment — a Psychologist's Look  | 50 |
| Семенов А. Л. Математическая информатика в школе<br>Semenov A. Mathematical Informatics at School  | 54 |
| Гадолина И. В., Константинова Т. Д., Пудиков А. Д.,<br>Репеченко В. Б., Семенов А. А., Юдина П. А.<br>Использование естественнонаучных дисков CD-ROM<br>в образовательном процессе<br>Gadolina I., Konstantinova T., Pudikov A., Repchenko V.,<br>Semenov A., Yudina P. Natural Science CD-ROM<br>in Educational Process | 59 |
| Болотова Н. В., Деревягина М. В., Ржевский Е. Ф.<br>Компьютер на уроках технологии (первый опыт и ближайшие<br>перспективы)<br>Bolotova N., Dereviagina M., Rzhvskiy E. Computer at the<br>Technology Lessons (the First Experience and the Nearest Perspectives)  | 61 |

**Директор**  
Кравцова А.Ю.  
**Зам. директора**  
Васильева Н.А.

**Ответственный секретарь**  
Иванова Т. В.

**Редакционная группа:**  
Васильев Б. М.,  
Кириченко И. Б.,  
Козырева Н. Ю.

**Компьютерная верстка**  
Кириченко И. Б.

**Оформление 1 и 2 сторон обложки — художник**  
Бажилин Э. В.

**Технический редактор**  
Луговская Т. В.

**Корректор**  
Антонова В. С.

**Экономический отдел**  
Бородаева З. В.

**Отдел подписки и распространения**  
Коптева С. А.  
(095) 208-30-78

**Информационное агентство ИА ИНФО**  
Факс: (095) 208-67-37  
E-Mail: info@tit-bit.msk.su

Нестеров В. М. Компьютерный мини-класс для слепых и слабовидящих детей  
Nesterov V. Computer Mini-Class for the Blind and Sight-impaired Children 63

#### МЕТОДИКА METHODICS

Залогова Л. А., Семакин И. Г. Уроки по Windows (продолжение)  
Zalogova L., Semakin I. Windows Lessons (continuation) 71

Грамолин В. В. Хочу работать в Windows! (Уроки работы в текстовом редакторе Word for Windows 6.0 русская версия)  
Gramolin V. I Want to Work with Windows! (Word for Windows 6.0 Russian Version Lessons) 81

Русakov С. В., Семакин И. Г. Базовый курс ОИВТ: «Пермская версия» (часть 7)  
Rusakov S., Semakin I. BCST Basic Course: "Perm Version" (Part 7) 98

Шафрин Ю. А. Чему учить школьников на уроках информатики?  
Shaphrin Y. What to Teach Pupils on CS Lesson? 105

#### APPLE ДЛЯ ОБРАЗОВАНИЯ APPLE FOR EDUCATIONS

Блинов А. Г. Программы для Macintosh  
Blinov A. Educational Software for Macintosh 109

Блинов А. Г. Комплект русскоязычного программного обеспечения на CD-ROM  
Blinov A. Russian Software Set on CD-ROM 115

#### ПРОФЕССИОНАЛЬНЫЙ КОМПЬЮТЕР PROFESSIONAL COMPUTER

Терновская Т.С. Вопросы маркетинга программных продуктов  
Terpnovskaya T. Software Marketing Problems 117

#### НОВЫЕ ПРОЕКТЫ NEW PROJECTS

Урнов В. А. Вдоль по дороге...  
Ournov V. Along the Road... 119

Белич Д. В. Сменим доску и мел на экран и лазер  
Belich D. Let's Change Black-Board and Chalk for Screen and Laser Beam 123

В подготовке материалов московского выпуска принимали участие  
А. Л. СЕМЕНОВ, Л. Б. ПЕРЕВЕРЗЕВ и П. А. ЯКУШКИН

Почту направлять по адресу: 103051, Москва, ул. Садовая-Сухаревская, д. 16, к. 9,  
журнал «Информатика и образование».

Телефон: (095) 208-30-78  
Факс: (095) 208-67-37  
E-Mail: info@tit-bit.msk.su

При перепечатке материалов ссылка на журнал обязательна. Редакция не несет  
ответственности за содержание рекламы и используемые в ней товарные знаки.

За содержание листингов программ редакция ответственности не несет.

Подписано в печать с оригинал-макета 28.08.95. Формат 70×100 1/16. Бумага офсетная № 1. Печать офсетная. Усл. печ. л. 10,40 + 0,32 вкл. Усл. кр.-отт. 11,70 + 1,30 вкл. Уч.-изд. л. 13,5 + 0,28 вкл. Тираж 11 845 экз. Заказ 1026.

Цена по подписке:  
для индивидуальных подписчиков 8 000 руб. (индекс 70423);  
для предприятий и организаций 25 000 руб. (индекс 73176).  
В розницу цена договорная.

Ордена Трудового Красного Знамени Чеховский полиграфический комбинат  
Комитета по печати и информации Российской Федерации. 142300, Чехов Московской обл.

# Семь пядей.

Представляем новую операционную Систему 7.5

Известно, что для работы на Макинтоше не нужно быть семи пядей во лбу.

Это действительно так. Макинтош сделал персональный компьютер понятным для самых неподготовленных пользователей; профессионалы же с удовольствием используют его, потому что дружелюбный интерфейс делает работу более эффективной и приятной.

Тем не менее, Apple Computer продолжает совершенствовать программное обеспечение Макинтоша.

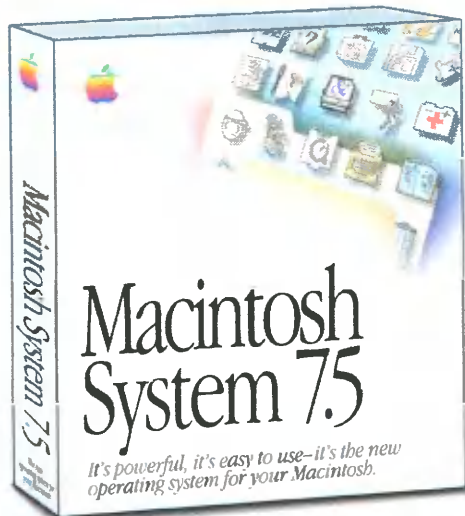
Операционная Система 7.5 содержит более 50 новшеств и

дополнений, делающих вашу работу еще проще. В их числе активная система помощи Apple Guide™ и AppleScript™ для автоматизации часто повторяющихся операций. Встроенная поддержка PC Exchange™ и Macintosh Easy Open упрощают обмен информацией между Макинтошем и IBM-совместимыми PC, в том числе с русской кодировкой. Улучшенное расширение QuickTime™

и Sound Manager 3.0 увеличивают производительность мультимедийных приложений, а функции Drag and Drop будут работать с картинками и звуковыми данными также, как они работали раньше с текстом.

Отдельного внимания заслуживает

встроенная в 7.5 графическая подсистема QuickDraw GX™ и сетевой интерфейс PowerTalk™ с функциями безопасности информации. Многие другие усовершенствования облегчат и упростят вашу работу на Макинтоше.



Русифицированная версия Системы 7.5 поставляется сегодня с каждой новой моделью Макинтоша. А те, кто уже работает на Макинтоше, могут приобрести новую версию системы у любого официального дилера Apple за небольшую плату. Чтобы узнать телефон ближайшего дилера, звоните по тел. (095) 978-8001.

Для этого вовсе не нужно быть семи пядей во лбу.

**ЛИНТЕХ**

# Virtual/PC

АО "ЛИНТЕХ"

119501 Москва, а/я 942

Тел/факс:

/095/ 273-50-14

E-mail:

shop @ lintech.msk.su

**Хватит мечтать - давайте действовать!  
Превратите КУВТ УКНЦ, "Корвет" и БК в IBM PC**

Принципиально новые системы "NET - RT11 & DOS - LINE" и "NET - CP/M & DOS - LINE" позволят Вам превратить КУВТ УКНЦ, "Корвет" и БК в классы IBM PC. На каждом рабочем месте Вы будете работать, как на IBM PC, под управлением MS DOS, использовать Norton Commander, Лексикон, Turbo Basic и другие популярные программы для IBM PC. При этом полностью сохраняется возможность использования всего существующего программного обеспечения для этих КУВТ.

Для модернизации КУВТ достаточно приобрести нашу систему и установить в КУВТ IBM - совместимый головной компьютер.

Локальные сети "NET - RT11 & DOS - LINE" и "NET - CP/M & DOS - LINE" объединяют с помощью высокоскоростных сетевых адаптеров в единое целое головной компьютер IBM PC и ученические машины. Скорость работы повышается в 30 - 100 раз, на каждом компьютере ученика обеспечивается полноценная работа без сбоев и зависаний благодаря отказу от использования стандартного сетевого оборудования и дисководов.

**Цена систем ниже цены одного IBM - совместимого компьютера. В настоящий момент ими оснащено более 600 компьютерных классов на территории России, Белоруссии, Украины и Казахстана.**

Все системы просты в установке и использовании, не требуют перемонтажа существующих линий связи, весь процесс модернизации стандартного класса занимает 2 - 3 часа. Гарантия - 3 года со дня приобретения.

**Министерство образования РФ рекомендует использовать системы "NET - RT11 & DOS - LINE" и "NET - CP/M & DOS - LINE" для модернизации КУВТ УКНЦ, "Корвет" и БК.**

