

# ИНФОРМАТИКА И ОБРАЗОВАНИЕ

6 1995



# Что такое Макинтош?



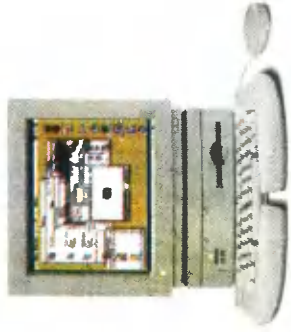
## Это Макинтош

Компьютер, удивительный по простоте обучения и использования. Для работы на нем не потребуются запоминать сложные команды и иметь специальную подготовку. Но самое главное - система работает на русском языке!



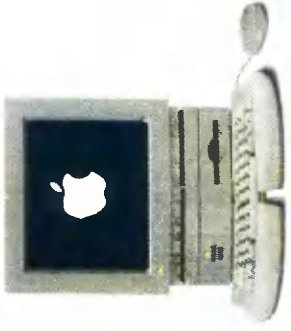
## Это опять Макинтош

Самый «студентоустойчивый», надежный и безопасный компьютер. Его устойчивость к перепадам напряжения и внешним воздействиям прекрасно сочетается с великолепной эргономикой и экологичностью.



## Это снова Макинтош

Богатейшая библиотека программ для образования, в том числе на русском языке, поможет прикоснуться к миру мультимедиа и свободно использовать цвет, звук и видео для создания собственных учебных курсов.



## Макинтош

Это компьютер, который уже более 10 лет работает в школах и университетах во всем мире. Он стал самым популярным компьютером в Америке и уверенно завоевывает сердца российских преподавателей и учащихся.

**Уважаемые читатели!**

**Поздравляем вас  
с Новым 1996 годом!**

*В 1995 году мы вместе отметили замечательный юбилей — 10-летие предмета «Основы информатики и вычислительной техники». Для нашей редакции 1996 год тоже юбилейный — журналу «Информатика и образование» исполняется 10 лет. (Первый номер вышел в августе 1986 года.)*

*За эти десять лет на страницах журнала были опубликованы дидактические материалы и методические приложения к учебникам информатики, методики преподавания информатики и передовой опыт учителей, нормативные документы по охране и оплате труда, стандартам образования, материалы по различным типам техники (IBM, Macintosh, УКНЦ, «Корвет», БК, «Ямаха» и др.), задачи выпускных и вступительных экзаменов и многое другое. В настоящее время журнал установил контакты с зарубежными изданиями аналогичного профиля и в 1996 году планирует публиковать материалы о зарубежном опыте использования информационных технологий в образовании. Мы продолжим информировать наших читателей о программах, конкурсах, выставках, конференциях, в которых смогут принять участие российские специалисты.*

*Благодарим всех наших читателей за отклики, пожелания, критику и надеемся, что и в Новом 1996 году мы снова будем вместе!*

**Поздравляем коллектив**

**Чеховского полиграфического комбината,  
который печатает наш журнал все эти годы,  
с тридцатилетним юбилеем.**

**Желаем нашим коллегам творческих успехов  
в реализации всех интересных планов и начинаний.**

# ОБЩИЕ ВОПРОСЫ

Л. Б. Переверзев,

заведующий лабораторией дизайна образования ИНТ

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

### Микромиры

#### Мир, не омраченный побочными вопросами

«Ребенок (а в действительности и большинство взрослых) живет в мире, в котором все понимается лишь отчасти: достаточно хорошо, но никогда достаточно полно» [2, 3].

Микромир, где «живет» черепашка, отличается тем, что все в нем может быть понято с гораздо большей, практически исчерпывающей полнотой. Ведь содержание этого микромира, само его становление и наполнение теми или иными объектами, системами, структурами и процессами целиком определяется действиями черепашки, которая лишь выполняет программы, составленные ребенком. (Мы отвлекаемся сейчас от анализа того, что происходит *внутри* компьютеров и что лишь частично понимается сегодня даже самыми передовыми учеными-компьютеристами.)

Инструментарий Лого непрерывно расширяется и совершенствуется. Новейшие его версии (например, пакет «ЛогоМиры» для компьютеров Macintosh) позволяют выводить на экран до ста отдельных черепашек, способных вступать в весьма сложные взаимодействия друг с другом. Выполняя составленные для них программы, они принимают любой заранее приготовленный для них облик, гибко трансформируя свою форму, цвет и сопутствующее звучание (в том числе речь и музыку, записанную пользователем через микрофон или скопированную с компакт-диска).

У ребенка есть возможность создавать множество различных микромиров Лого, наблюдать происходящие в них события, до малейших деталей исследовать логику их взаимосвязи и управляющие ими законы, равно как и произвольно менять последние. «Дети узнают, на что похоже изучение свойств избранного ими и не омраченного побочными

вопросами мира. Поступая так, они учатся переносить исследовательские привычки из собственной жизни на построение формальной области научной теории» [2, 3].

Вспомним уже знакомый нам микромир геометрии, создаваемый ребенком посредством черепашки-чертежника и состоящий из точек, линий и углов. Добавив туда *часы* (неотъемлемую часть каждого компьютера) и введя дополнительные правила и команды, устанавливающие зависимость пространственного положения и направления черепашки (теперь уже только дисплейной) от нами же приписанной ей массы, приложенной силы и истекшего времени, мы получаем микромир, позволяющий строить наглядные модели механического движения.

Дисплейную черепашку, принимаемую нами теперь за *частицу*, обладающую массой и скоростью, именуют Диначерепашкой, и занятия с ней открывают множество новых возможностей для экспериментального постижения детьми теоретических основ физической науки. Общий подход остается прежним: конструктивное изучение сложных, странных и противоречащих повседневной практике научных понятий (скажем, законов небесной механики) начинать с того, что хорошо согласуется с уже имеющимся у ребенка субъективным опытом. Сравним его еще раз с тем, что обычно имеет место в «инструктивном» преподавании.

### Приобщиться к размышлениям Ньютона

Поскольку традиционная педагогика не может «предъявить» ньютоново движение чувственному восприятию учащихся, она вынуждена объяснять его в форме отвлеченно-опосредованного и крайне математизированного описания. «Это движение по большей части изучается через преобразование уравнений, но не через манипулирование объектами» [2, 3]:

\* Продолжение. Начало см.: Информатика и образование. 1995. № 5.

ученик сперва должен потратить немало часов (если не лет), чтобы научиться работать с самими уравнениями и лишь потом использовать их как модель механической вселенной. Оттого-то данный раздел физики считается доступным лишь школьникам средних классов, прошедшим долгий путь штудирования его формально-алгебраических предпосылок.

При наличии компьютерного микромира физики нет нужды ждать годы, пока ребенок овладеет нужными уравнениями. Ведя диалог и взаимодействуя с Диначерепашкой, даже малыши легко знакомятся с общим характером движения по Ньютону. Конечно, для настоящего его понимания учащемуся не обойтись без главного инструмента концептуализации и теоретического овладения таким микромиром — дифференциального исчисления. Поэтому все базовые программы, определяющие поведение Диначерепашки, «обеспечивают формализацию, благодаря которой мы можем овладеть нашей быстро ускользающей мыслью».

Идя обходным путем (через арифметику, алгебру, тригонометрию, математический анализ) к этой формализации, мы лишь в своем варианте повторяем путь от собственных записей Ньютона к записям, представленным в современных учебниках» [2, 3]. А это, в свой черед, «позволяет учащимся более непосредственно приобщиться к тому, о чем размышлял Ньютон, прежде чем начал записывать свои уравнения» [2, 3].

Ребенок также волен задумывать микромир, иллюстрирующий логическую «правильность» не только идей Ньютона, но и многих других — «таких, как исторически и психологически значимые идеи Аристотеля, более сложные идеи Эйнштейна и даже «обобщенный закон движения в мире», в рамках которого действуют бесконечно разнообразные законы движения, которые индивиды могут придумывать для себя». Располагая подобными микромирами, «учащиеся в соответствии со своими желаниями могли бы продвигаться от Аристотеля к Ньютону и дальше к Эйнштейну через множество промежуточных миров» [2, 3].

### Овладение динамикой

Задержимся еще немного на микромирах классической механики. Первые два закона Ньютона усваиваются детьми путем создания череды микромиров, где состояние Диначерепашки сначала задается положением и скоростью, а затем специальной командой «изменяем скорость на  $x$ , не обращая внимания на то, какая при этом получается скорость». Но у школьников больше всего затруднений вызывает третий закон, согласно которому Вселенная трактуется как механизм, движущийся благодаря воздействию частиц друг на друга без какого-либо внешнего вмешательства. Что-

бы смоделировать подобную Вселенную в Лого, нужно иметь уже не одну, а множество черепашек, *взаимодействующих между собой*.

Сразу это сделать довольно сложно. Сперва создается микромир, где пары «зеркально связанных» черепашек, еще не имеющих ни массы, ни скорости, *отдают команды друг другу* согласно правилу: «Всегда, когда отдается команда ВПЕРЕД (или НАЗАД), обе черепашки проделывают одно и то же; всегда, когда отдается команда ПОВЕРНУТЬ НАПРАВО (или ПОВЕРНУТЬ НАЛЕВО), одна черепашка делает обратное тому, что делает другая». Иными словами, при поворотах каждая черепашка ведет себя симметрично своему партнеру, как бы отражая его движения в зеркале.

В следующем микромире зеркальная связь применяется к черепашкам-частицам, уже наделенным скоростью и способным приобретать ускорение. «Ни одно из статичных изображений не в состоянии воспроизвести то зрительное возбуждение, которое вызывает этот калейдоскоп динамических изображений, когда светящиеся цветные точки образуют кружащиеся в легком танце пары. Конечный продукт волнует так же, как производные искусства, но процесс, результатом которого он является, включает умение мыслить в соответствии с действиями и реакциями связанных движущихся объектов» [2, 3].

Применяя команды УСТАНОВИТЬ СКОРОСТЬ и ИЗМЕНЯЕМ СКОРОСТЬ, ребенок учится приводить в движение не только самих черепашек, но и то, что он с их помощью начертил (дисплейные черепашки в новых версиях Лого способны сами принимать облик запрограммированной или нарисованной учащимися, а также транспортированной им извне фигуры, фотоизображения и т. п.). У него рождается интерес к придумыванию все более сложных заданий по изменению формы и размеров движущихся на экране объектов, ему становится подвластной динамика и открывается множество увлекательных возможностей для ее экспериментального исследования. Все это как небо от земли отличается от происходящего на уроках физики в традиционной школе.

В обычной школьной лаборатории перед детьми ставится задача установить данную истину. В лучшем случае они осваивают, что «это и есть единственный способ общения с миром» [2, 3]. В динамичных микромирах черепашки дети приходят к пониманию истины иного рода — *почему* эти миры работают так, как они работают. «Проверяя множество различных законов движения, дети обнаруживают, что законы Ньютона действительно наиболее экономичный и элегантный способ описания движущихся вокруг объектов» [2, 3]. Построение же множества собственных микровселенных «играет роль трамплина, позволяющего совершить прыжок от элементар-

ного понимания ньютоновой механики к пониманию движения планет и принципов разработки космических проектов» [2, 3].

### Инкубаторы знаний

Таким образом, роль микромиров не исчерпывается задачами моделирования одной лишь «внешней» реальности. До сих пор они рассматривались как инструмент, помогающий решать педагогическую проблему *предпосылок*, возникающую при структурировании нового учебного содержания — абстрактных теоретических понятий из области математики и физики. «Но микромиры являются также способом разрешения другого сорта проблем, относящихся не к системе знаний, а к самому индивиду. Проблема состоит в нахождении контекста для построения «ошибочных» (или, точнее, переходных) теорий. Каждый из нас учится, конструируя и используя мир и строя теорию, но по большей части на такой теории мы лишь оттачиваем свой ум, и как теория она нужна нам лишь для того, чтобы позднее от нее отказаться» [2, 3].

Пиаже особенно настаивал на том, что «необычные теории маленьких детей нельзя характеризовать как недостаток или когнитивный просчет, эти теории служат для наращивания когнитивных мускулов, для формирования и использования навыков, необходимых для построения более общепринятых теорий» [2, 3]. Микромиры, таким образом, оказываются «инкубаторами знаний», или «местом для выращивания специфического вида плодотворных идей или интеллектуальных структур» [2, 3]. Дети, их создающие, «учатся математике и естественным наукам в среде обучения, в которой истина и ложь не являются решающими критериями.

Подобно учебе в хорошей художественной школе, дети осваивают технические знания как *средства* для получения творческого и лично значимого результата... Истинность и ложность теории вторична по сравнению с ее вкладом в учение детей» [2, 3]. К сожалению, высказанные выше идеи все еще довольно медленно завоевывают себе сторонников среди педагогов. Большинство из них просто не имеет соответствующей подготовки и чувствует себя крайне неуверенно при встрече с компьютером, а кое-кто даже считает, что использование последних для целей образования приведет к еще большей «механизации» мышления современного человека. Не превратится ли при этом ребенок в машину?

### Мышление и компьютер

«Я вполне осознаю захватывающую власть компьютеров, работающих в диалоговом режиме, и воздействие этих устройств на наши представления о самих себе, когда они берутся в качестве модели, — соглашается Пейперт [2, 3]. Именно поэтому вся работа по

созданию Лого есть попытка направить это воздействие в позитивное русло.

Например, очень многих пугает распространившаяся по всему миру эпидемия компьютерных игр. Вместо того чтобы с порога их отвергать, как безусловное зло, школа вполне могла бы обратить их себе на пользу. «Работая над Лого, мы создавали варианты устройств, в которых плодотворные идеи физики, математики или лингвистики включены таким образом, что играющие на них дети усваивают эти идеи наиболее естественным для себя способом, похожим на тот, каким люди учатся говорить. Захватывающая власть компьютера, которой так опасаются критики, превращается в полезное орудие обучения» [2, 3].

Далее, когда ребенок сам «может заставить компьютер создавать мультипликацию, у него появляется новое, личностное отношение к тому, что он видит на экране телевизора или игрового автомата. Динамические зрительные эффекты, которые мы наблюдаем, смотря телевизор, мультипликационный фильм или стреляя в видеоигре, побуждают нас задаться вопросом: «Как было достигнуто то, что мы наблюдаем?» [2, 3] Поиск ответа ведется на путях рационального исследования, укрепляющего критико-аналитический и конструктивный подходы ко всему, пробуждающему наше любопытство.

Наконец, всячески поощряется тот же подход к самому процессу познавательной деятельности. Уже не раз отмечалось непрерывное участие в нем рефлексии и стремление воспитать у ребенка неизменную привычку быть исследователем собственного мышления. Предложение «стать в позицию черепашки» и отождествиться с нею, чтобы найти решение трудной задачи, отнюдь не поощряется на «человечность» нашего интеллекта. Совсем наоборот.

Не только не страшно, но в ряде случаев чрезвычайно кстати сознательно мыслить по аналогии с компьютером. Во-первых, так обеспечивается прочное закрепление ряда операционально-мыслительных навыков, необходимых при усвоении некоторых важных разделов математики и грамматики. Во-вторых, «благодаря сознательному умению имитировать механическое мышление учащийся получает возможность отличить механическое мышление от мышления, таковым не являющимся. Упражняясь в различении одного и другого, ребенок разовьет свою способность выбирать тот стиль когнитивного мышления, который наиболее подходит для решения данной проблемы», что ведет «к интеллектуальному совершенствованию на новом уровне» [2, 3].

Более того, развитию естественного человеческого интеллекта путем работы в компьютерных микромирах весьма помогает знакомство с теориями интеллекта искусственного, т. е. машинного. Одна из них, разрабатываемая Пейпертом и Марвином Минским, од-

ним из создателей современной теории автоматов, существенно перекликается с философией Лого.

### Сообщество разума

Теория Минского—Пейперта гласит, что абстрактная универсальная машина, обладающая чем-то, схожим с интеллектом человека, строится из огромного количества узкоспециализированных автоматов низшего ранга, или *агентов*. Каждый агент сам по себе лишен какого-либо интеллекта и предназначен для выполнения лишь одной, строго определенной и достаточно простой операции. Любая сложная задача разбивается на подзадачи более простого вида, а те, в свою очередь, на еще более простые — и так до тех пор, пока исходная задача не будет представлена связанной совокупностью настолько простых «атомарных» заданий, что с ними могут справиться абсолютно «несмышленные» агенты.

«Интеллектуальность» же такого рода абстрактной машины, известной под названием *The Society of Mind* (Сообщество разума), заключается в программе «общения» и продуктивного «сотрудничества» агентов в зависимости от конкретных обстоятельств: она предполагает, с одной стороны, анализ и распределение заданий между агентами, а с другой — организацию их совместного функционирования с установлением среди них относительного старшинства и определенных привилегий.

Означенную теорию допустимо применить для более отчетливого понимания процессов, происходящих в уме почти всех детей четырех или пяти лет от роду, когда они сталкиваются в экспериментах Пиаже с так называемым *вопросом о сохранении*. Многочисленные лабораторные наблюдения показывают: дети этого возраста единодушно утверждают, что количество жидкости уменьшается при переливании ее из узкого и высокого сосуда в широкий и низкий (и наоборот). А года два-три спустя те же дети меняют свое мышление и столь же единодушно настаивают, что, как ни переливай жидкость, ее остается ровно столько же, сколько было и раньше. Чем это объяснить?

Согласно Пиаже, у детей именно в этом возрасте появляются необходимые для правильного суждения интеллектуальные структуры, возникающие естественно-органически и абсолютно независимо от воздействия внешнего окружения. Традиционное психологическое истолкование сводит это к преобладанию «внешних впечатлений»: разум маленького ребенка направляется восприятием и просто отражается считаться с тем, что вещи не таковы, какими кажутся. Объяснение, опирающееся на модель «Сообщества разума», принципиально иное.

«Будем исходить из существования в уме

ребенка трех агентов... Первый агент,  $A_{\text{высота}}$ , судит о количестве жидкости и о всем остальном по вертикальной шкале... это агент из практической жизни ребенка» [2, 3], и он судит по принципу: «все, что выше, то больше». Второй агент,  $A_{\text{ширина}}$ , судит о количестве по горизонтальной шкале и вступает, «когда приходится решать, что в море очень много воды», но для ребенка этот агент менее действителен, чем  $A_{\text{высота}}$ . Третий агент,  $A_{\text{история}}$ , сообщает, что количество осталось тем же самым, поскольку с ним ничего не произошло.

«В эксперименте с детьми, не соблюдающими принцип сохранения, каждый из трех агентов принимает собственное решение и поднятую ими суматоху необходимо утихомирить». Обычно «голос  $A_{\text{высота}}$  звучит громче всех. Но как только его громкость снижается, ребенок переходит на следующую стадию своего интеллектуального развития» [2, 3]. Однако происходит это не только по чисто органическим причинам. Принципиально мыслимы «три способа, изменяющие степень участия этих агентов.

$A_{\text{высота}}$  и  $A_{\text{ширина}}$  могли бы «усовершенствоваться», если бы, например,  $A_{\text{высота}}$  принимался бы во внимание только в тех случаях, когда предметы ничем не отличались бы, кроме высоты» (т. е. имели бы одинаковое поперечное сечение). Второй способ — «изменить «старшинство» в привилегиях и отдать решающий голос  $A_{\text{история}}$ . Ни один из этих умозрительных способов невозможен в реальности. Но существует третий способ, приводящий к тому же эффекту более простым путем. Он в том, чтобы нейтрализовать голоса  $A_{\text{высота}}$  и  $A_{\text{ширина}}$ , приведя их в столкновение друг с другом и добавив к ним еще одного агента,  $A_{\text{геом}}$ , действующего как их наставник.

Когда  $A_{\text{высота}}$  и  $A_{\text{ширина}}$  согласуются между собой,  $A_{\text{геом}}$  опирается на их показания, как на весьма авторитетные. Если же они рассогласуются между собой,  $A_{\text{геом}}$  определяет это, и их голоса нейтрализуются. При всем том « $A_{\text{геом}}$  вовсе не должен «понимать» причин решения, принимаемого относительно  $A_{\text{высота}}$  и  $A_{\text{ширина}}$ .  $A_{\text{геом}}$  не знает ничего, кроме того, согласуются ли эти агенты, и если согласуются, то в каком направлении» [2, 3]. (Не исключено, что  $A_{\text{геом}}$  подсказывается ребенку образом действий отца, не всегда успешно справляющегося с налаживанием порядка в семье.)

Разумеется, абсурдно утверждать, что происходящее в уме ребенка, соблюдающего принцип сохранения, может быть объяснено столь простой моделью с четырьмя агентами — их здесь требуется десятки и сотни. Но при всей своей упрощенности модель точно передает главный принцип рассматриваемой теории интеллекта, будь то естественного или машинного: «компоненты данной системы более напоминают людей, чем логические посылки, и взаимодействие этих компонентов

скорее напоминает социальное взаимодействие, чем операции математической логики... В частности, мы можем трактовать овладение логикой как непрерывное овладение социальным и личностным в познании» [2, 3].

### Уравнение в лицах

Внешне парадоксален, но глубоко закономерен тот факт, что наиболее адекватной моделью любой «интеллектуальной машины», создаваемой ребенком в микромирах Лого, выступает человек. Причем не какой-нибудь изолированный «абстрактный индивид», но человек *общественный*, включенный в сеть социальных и культурных связей с другими людьми. Для человеческого же интеллекта первичным двигателем служат волнующие его аффекты, а проводником в поисках — эстетическая ориентация. Даже простейшие ее типы существенно повышают эффективность учения в областях знания, считающихся царством бесстрастной логической строгости.

Группе студентов-нематематиков, вооруженных лишь карандашом и бумагой, предложили найти доказательство теоремы о том, что корень из двух является иррациональным числом. Оказалось, что «люди с весьма скромным запасом математических знаний в состоянии решить эту задачу, если они работают в условиях эмоционального подъема, побуждающего их к поиску решения, даже если им не хватает математических знаний» [2, 3]. Откуда взялся этот эмоциональный подъем? Почти все испытуемые прошли через весьма примечательный этап поиска решения, смысл которого легче всего понять, вообразив себя на их месте.

Составив исходное уравнение  $\sqrt{2} = p/q$ , где  $p$  и  $q$  — целые числа, «предположим, что мы действительно не верим, что 2 может быть представлен таким образом. Обоснуем нашу уверенность, выявив какую-нибудь несуразность, а фактически противоречие, скрывающееся за этим непроницаемо невинным поверхностным впечатлением» — займемся «взаимодействием скрытого и явного содержания» [2, 3].

Чувствуя наличие в данном уравнении некоего внутреннего противоречия, все начали пробовать различные способы его преобразования, идя по пути математических «свободных ассоциаций», хотя никто ясно не предвидел, куда это может его завести. И вот в какой-то момент, получив после ряда преобразований  $p^2 = 2q^2$ , они начинали испытывать особенно сильное волнение. Почему?

«Насколько можно было судить... испытуемые почти не интересовались не только полученными ими самими, но даже и исходным уравнением» [2, 3]. Эмоциональная реакция менялась от испытуемого к испытуемому и, как выяснилось, у каждого вызывалась несколькими причинами. Одни «явно задава-

лись целью избавиться от квадратного корня, другие такой цели не ставили, но тем не менее радовались» [2, 3], когда им это удавалось. По аналогии с уже упомянутой моделью «Сообщества разума», суть происходящих при этом мыслительных процессов допустимо выразить следующими двумя описаниями.

При первом «исходное уравнение можно представить в виде ситуации с тремя действующими лицами, среди которых главным, или «испытуемым», лицом является  $\sqrt{2}$ . Другие два действующие лица,  $p$  и  $q$ , — это статисты, высказывающие нечто об «испытуемом» лице. Когда исходная ситуация преобразуется в  $p^2 = 2q^2$ , то соотношение действующих лиц резко меняется, как в случае восприятия младенцем игры в прятки, когда произносимое взрослым «ку-ку» помогает ребенку отыскать взглядом лицо играющего с ним человека. Теперь  $p$  становится «испытуемым» лицом, а  $\sqrt{2}$  исчезает. Не связано ли возникающее при этом удовольствие с тем же источником, который заставляет радоваться младенца, слышащего от взрослого: «Ку-ку?»

При другом описании удовольствие связывается с наблюдением того, что  $\sqrt{2}$  исчезает не бесследно. В уравнении  $p^2 = 2q^2$  присутствует 2! Однако два эти вхождения 2 в уравнения различаются по своим ролям примерно также, как различается роль слова в каламбуре и в «серьезной» речи. Направивается и параллель с фрейдовской «конденсацией», как фундаментом остроумия, когда «абстрактные» описания указывают на самые различные «конкретные» вещи. «Не позволяет ли нам это предположить, что математика имеет гораздо больше общего с шуткой, мечтой и историей, чем это обычно признается?» [2, 3]. Если дело обстоит именно так, то в плане математики очень важно понять, как эстетическая, шутливо-игровая сторона математики соотносится с ее «серьезным», логическим, или функциональным аспектом.

### Эстетическое и функциональное: взаимная эксплуатация

Помня, что получение  $p^2 = 2q^2$  и работа с ним производятся всё же не только ради удовольствия, постараемся превратить это уравнение в подцель главной цели — доказательства того, что  $\sqrt{2}$  — иррациональное число. «Как же нам объединить функциональное с эстетическим?»

Функциональная система «как перводвигатель запускает универсум суждений, относительно которых могут формулироваться подцели» [2, 3]. Степень важности, или приоритетности последних зависит от степени «влиятельности», приписываемой тому «действующему лицу», с которым связывается их выдвижение. Например,  $p$  может быть повышен в субординации, из статиста переведен в солиста. Внутри соответствующей системы



ситуационных ограничений этот перевод явит собою правильное определение подцели «найти числовое решение уравнения».

С другой стороны, и главная цель имеет «нечто общее с нематематическими ситуациями в жизни и в художественной литературе» [2, 3]. В своем крайнем выражении данный способ осмысления «заставляет нас видеть в математике (и даже в ее частностях) нечто выходящее за пределы этой науки. Хотя действующими лицами остаются математические объекты, разворачивающийся сюжет описывается в новых терминах. Даже в менее экстремальных случаях можно пронаблюдать, как эстетическое и функциональное начала вступают в символическую связь, так сказать, взаимно эксплуатируют друг друга. Математическая функциональная цель достигается через игру с подцелями, сформулированными в иных, нематематических способах рассуждений, с подцелями, вытекающими из соответствующих нематематических способов познания. Таким образом, функциональное эксплуатирует эстетическое. Но в той же степени, в какой мы рассматриваем (в данном случае, почти следуя Фрейдю с его теорией подсознательного) математические процессы как находящиеся под воздействием внешних им процессов, истинно и обратное» [2, 3].

Так разрешение проблемы с  $\sqrt{2} = p/q$  достижимо как по способу классического, приписываемого еще Евклиду, «пошагового рассуждения» (его можно называть также «серийным»), так и «гештальта», или «единоразового озарения». Давайте сопоставим их между собой.

### Бодрящее дух упражнение собственных возможностей

При первом способе «из уравнения  $p^2 = 2q^2$  следует, что  $p^2$  — четное число, следовательно,  $p$  также является четным числом. По определению четного числа это значит, что  $p$  в два раза превышает какое-то целое число, которое мы обозначим как  $g$ . Итак:

$$p = 2g,$$

$$p^2 = 4g^2,$$

$$2q^2 = 4g^2, \text{ напомним, что } p^2 = 2q^2 (!),$$

$q^2 = 2g^2$ , а это значит, что  $q$  — также четное число, что по меньшей мере странно, поскольку первоначально мы выбрали  $p$  и  $q$  таким образом, чтобы они не имели общего делителя. Тем самым мы получили противоречие» [2, 3].

Второй — «гештальтный» — способ «связан с определенным восприятием целых чисел как выражаемых через ряд множителей, каждый из которых является простым числом... Если вы твердо усвоили этот способ выражения целого числа, то, возможно, вам нетрудно понять, что квадрат целого числа всегда представим в виде ряда четных чисел простых множителей». А тогда «вам столь

же нетрудно понять абсурдность выражения  $p^2 = 2q^2$ , поскольку ряд, состоящий из четного числа простых множителей, приравнивается к ряду ( $q^2$  плюс добавочный множитель 2) с нечетным числом множителей, также представляющих собой простые числа» [2, 3].

Какое из двух доказательств математически предпочтительнее?

Гештальтное, или непосредственно-образное, — остроумно, лаконично, изящно. Но и пошаговое, или серийное, по-своему очень привлекательно. Есть «нечто весьма плодотворное в том способе, каким мы ухватываем и неумолимо проводим серийный процесс. Я не просто осознаю риторическую убедительность своих доводов для другого человека, хотя подобная убедительность является важным фактором для зрелищной стороны математики. Скорее, я осознаю, что требуются очень небольшие познания в математике для пошагового продвижения, но, как только вы начали свой путь, вы легко построите все доказательство» [2, 3].

И тот, и другой подход неизбежно оказывается эмоционально окрашенным. «Для кого-то такой процесс ассоциируется с временным подчинением. Кто-то может воспринять его как отказ от себя в пользу математики или другого человека или как чистящую себя в другом. Но для кого-то такой процесс никакое не подчинение, а бодрящее дух упражнение собственных возможностей» [2, 3]. В конечном счете на первый план вновь и вновь выходит одна и та же проблема: соотношение того, что происходит в изучаемом предмете, с тем, что происходит в учащемся человеке. Решить эту проблему можно, лишь став на позиции антропологизма в образовании.

### Антропологический принцип и «школа самбы»

Антропологизм — философское направление, видящее в человеке микрокосм, малую вселенную и модельный прообраз всего, что можно обнаружить и создать в макрокосме, т. е. Большой Вселенной. Самба — традиционный бразильский танец и импровизационный музыкальный жанр, возникший из сплава португальских мелодий и африканских ритмов. Что между ними общего?

Пейперт объединяет их в рабочей метафоре, призванной наглядно иллюстрировать суть образовательной доктрины Лого. В соответствии с этой доктриной учебная среда возникает как естественное продолжение целостной жизненной практики. Последняя же находит наиболее концентрированное выражение в тех видах художественной активности, которые символизируют социальные и культурные ценности, близкие и дорогие всем членам конкретного человеческого сообщества.

Частичным аналогом такой учебной среды служат бразильские «школы самбы» —

множество неформальных клубов, где любительские группы готовятся под руководством квалифицированных специалистов к ежегодному карнавалу в Рио-де-Жанейро. Его сердцем является двенадцатичасовая процессия, в ходе которой десятки и сотни конкурирующих групп выступают с музыкой, песнями, танцами и уличными представлениями на темы национальной истории и фольклора вперемежку с политической сатирой на злобу дня.

В течение года каждая группа самбы, куда входят люди всех возрастов (от внуков до бабушек и дедушек), определяет тематику своего будущего представления, сочиняет тексты, разучивает вокальные и хореографические номера. То, чему обучают в такой «школе», происходит естественно, но вполне продуманно. Опытные мастера демонстрируют художественные приемы, дети их внимательно наблюдают и репетируют, а потом присоединяются к взрослым.

Среда Лого похожа на «школу самбы» прежде всего тем, что математика, как деятельность, объединяет там и школьников, и учителей. «Эта деятельность столь изменчива и столь богата открытиями, что уже в первый день занятий программированием ученик способен сделать нечто новое и поразить учителя» [2, 3]. Так, по теме обязательного школьного курса это может быть изучение свойств треугольников или дробей. Но способы, с помощью которых тот или иной ребенок может прийти к правильному решению в среде Лого, могут оказаться совсем различными, а подчас довольно необычными и оригинальными.

Различие состоит в том, что «школы самбы» «корнями уходят в народную культуру» и «осваиваемые там знания неразрывно связаны с ней» [2, 3]. Среда Лого — это «искусственный оазис, в котором люди сталкиваются со

знаниями (математическими и математическими), но они отделены от основного потока окружающей культуры, а на самом деле противоположны ценностям, утверждаемым в такой культуре» [2, 3]. Ведь «школы самбы» не изобретались исследователями и не учреждались по приказу правительства, а всё же возникли. Так же и новые формы учебных ассоциаций, применяющих Лого, должны своими корнями уходить в культуру, а не возникать по предписанию чиновников.

Отсюда обращающийся к Лого «педагог обязан придерживаться концепции антропологизма» и «научиться разбираться в том, какие элементы культуры влияют на интеллектуальное развитие. Особенно важно следить за возникающими там тенденциями, ибо лишь с учетом последних есть шанс реально видоизменить сложившийся образовательный процесс» [2, 3]. Все «новаторы в образовании должны осознать, что для успешного осуществления их предложений необходимо, чтобы эти предложения находили отклик в окружающей культуре, и в них следует использовать динамичные тенденции культуры в качестве посредников педагогического вмешательства» [2, 3].

Одна из таких тенденций — массовое проникновение новых технологий во все сферы жизни; вторая — столь же массовое разочарование школой. Обе тенденции можно объединить таким образом, чтобы это принесло пользу детям, родителям и учению. Их слияние дает надежду на возникновение компьютерной учебной среды, которая была бы способна стать альтернативой традиционному классу и обучению и которую заслуженно можно было бы называть «школой самбы по программированию».

## Литература

1. Сопрунов 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. Пеўнерт 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.

## МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

## ИНФОРМАЦИОННОЕ ПИСЬМО

12.09.95

Москва

№ 42/3

**ОБ ИНФОРМАЦИОННОЙ КУЛЬТУРЕ  
В СЕМЕЙНОМ ОБРАЗОВАНИИ  
(Письмо 1)****1. Положение дел в сфере информационной культуры  
семейного образования**

В настоящее время происходит активное внедрение чисто развлекательной культуры в повседневный быт детей и взрослых. Это относится ко всем сферам культуры — книги, музыка, видео, а также компьютерная культура, предлагаемая игровыми телевизионными приставками и большинством коммерчески распространяемых игр для компьютеров. Это — игры «активных действий», т. е. «бегалки и стрелялки», изредка имитаторы автомобилей, самолетов и др.

Эти игры не поощряют творческую мотивацию поведения. Более того, поскольку они моделируют такую деятельность, в которой результат достигается с помощью агрессивных приемов, постольку ребенок привыкает интересоваться лишь теми занятиями, где результат внешне эффектен и доступен. Тем самым дети отучаются от занятий, требующих постоянных интеллектуальных усилий, т. е. именно от тех, которые носят развивающий и обучающий характер.

Игры, в которых средством достижения успеха является агрессивное поведение игрока, естественно, формируют агрессивную направленность личности ребенка. Происходит отчуждение ребенка от родителей и сверстников. Формируется асоциальное поколение с агрессивным мышлением и умением (и стремлением) действовать в ситуациях, где выигрыш демонстративен и легко достигается. Механизм повышенной криминализации такого поколения очевиден.

Общий вывод: информационная культура детей в российских семьях развивается в направлении ухода от действительности (аутизации), тем более что трудности жизни многих семей этому способствуют. Происходит духовная экспансия агрессивных образцов поведения в «экологию информационного пространства семьи». Это требует принятия активных мер со стороны государства, и в первую очередь учреждений системы Министерства образования.

**2. Программа действий Министерства образования по развитию  
информационной культуры семейного образования**

Министерством образования России поднимается вопрос о расширении комплекса технических средств, а также средств информации и коммуникации, позволяющих сформировать многокомпонентную информационную культуру семейного образования. (Информационно-методическое письмо Министерства образования № 5/11 от 04.01.95 г.)

Развитие гуманистического творческого потенциала подрастающего поколения предполагает осуществление следующей программы действий:

- разработка, производство и распространение гуманистических интерактивных средств и технологий информационных культур в школьном и семейном образовании;
- дифференцирование предлагаемых средств информационной культуры с учетом существующего разброса покупательных возможностей семей разных слоев общества;
- создание системы пропаганды и распространения гуманистической семейной информационной культуры;
- разработка государственных стандартов развития семейного интеллектуального досуга и семейной информационной культуры, опирающихся как на государственное, так и внебюджетное финансирование.

### 3. Пропаганда информационной культуры семьи

Для того чтобы в повседневную жизнь российских семей внедрялась не чисто развлекательная, а гуманистическая образовательная, интеллектуальная культура, нужна ее активная пропаганда.

Важно, что сегодня имеются благоприятные предпосылки для распространения такой культуры. Действительно, стало взрослым поколение, детство и юность которого прошли в условиях, когда был высок престиж интеллектуальных занятий. Это поколение само стало растить детей. В этих семьях родители ностальгически понимают, что детский интеллект необходимо активно развивать, и именно на такие семьи можно и нужно ориентироваться в первую очередь.

Для широкой пропаганды информационной культуры в семейном и школьном образовании необходимо также создание сети центров пропаганды информационной культуры семьи.

Такие хозрасчетные центры Министерство рекомендует создавать на базе Домов детского творчества, детских библиотек, кинотеатров, музеев, в особенности музеев материальной культуры, школ.

Особое внимание следует уделить создающимся в настоящее время медиациентам и информационно-прокатным центрам. Медиациенты призваны внедрять новые информационные технологии, а также традиционные средства обучения для обеспечения разнообразных образовательных услуг, в том числе помощь родителям и их детям при новых формах получения образования – семейном образовании и экстернате и т. п. Им легче других взять на себя пропаганду (и распространение) развивающих и обучающих средств, не только компьютерных, и стать региональными информационно-пропагандистскими центрами.

Более подробную информацию и рекомендации по созданию таких центров и другую помощь можно получить в Министерстве образования (Голосов А. И., тел. : (095) 925-20-90).

Заместитель федерального Министра

А. Г. Асмолов

#### УВАЖАЕМЫЕ ЧИТАТЕЛИ!

Редакция журнала «Информатика и образование» сообщает об изменении своих банковских реквизитов.

*Расчетный счет для Москвы и Московской области:*

800345203 в отделении «Чеховское» АБ «Торибанк»,  
МФО 996095, уч. ЕМ.

*Расчетный счет для России и стран СНГ:*

800345203 в отделении «Чеховское» АБ «Торибанк»,  
корр. счет № 715161800 в ЦРКЦ ГУ ЦБ РФ, МФО 201791.

Обращаем внимание наших читателей, что журналы, оплата за которые переведена на старый счет (филиал ММКБ «Интеллект»), будут высылаются редакцией только при условии прихода денег до 1 января 1996 года.

## МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

## ИНФОРМАЦИОННОЕ ПИСЬМО

12.09.95

Москва

№ 43/3

**ОБ ИНФОРМАЦИОННОЙ КУЛЬТУРЕ  
В СЕМЕЙНОМ ОБРАЗОВАНИИ  
(Письмо 2)****Опыт пропаганды и распространения  
информационной культуры семейного образования**

В письме Министерства образования «Об информационной культуре семейного образования», вып. 1 сформулирована программа борьбы с аутизацией детей, которая вызывается развлекательной культурой и агрессивными компьютерными играми. Там же предложены конкретные шаги по созданию центров пропаганды гуманистической информационной культуры семейного образования.

В этой связи представляет интерес опыт пропаганды интеллектуальной атмосферы семьи, накопленный промышленно-коммерческим центром «Компьютеры для населения», функционирующим в рамках договора о творческом сотрудничестве с Министерством образования Российской Федерации.

**1. Выставка домашних компьютеров**

Центр «Компьютеры для населения» совместно с Политехническим музеем в Москве организовал выставку-консультацию-продажу «Домашние персональные компьютеры для детей и взрослых», где предлагается широкий спектр персональных компьютеров — от простых моделей для детей до сложных профессиональных. Важно, что в соответствии с пожеланиями Министерства (см. письмо 1, п. 2) учитывается существующий разброс покупательных возможностей семей: цены на такие компьютеры лежат в диапазоне от 3 до 150 минимальных зарплат. На выставке желающие могут бесплатно поиграть на домашнем компьютере и получить квалифицированную консультацию по выбору оптимальной модели такого компьютера с учетом доходов семьи, а также игровой или творческой мотивации ребенка.

**2. Программное обеспечение домашних компьютеров**

Подход предприятия «Компьютеры для населения» к проблеме семейной информационной культуры позволяет дать практически любой семье действующий компьютер.

Важно, что там же семьям предлагается программное обеспечение домашних, в первую очередь детских, компьютеров — начиная от игр различной степени сложности (активных, диалоговых, логических, стратегических) и кончая текстовыми, графическими и музыкальными редакторами, базами данных, а также возможностями для самостоятельного создания программ. Таким образом, ребенок получает средства не только для игры, но также и для обучения и самовыражения с помощью компьютерной техники.

По предложению Министерства образования центр начал сбор, анализ и систематизацию обучающих и развивающих компьютерных программ для школьного и семейного образования, в том числе для IBM-совместимых компьютеров. По просьбе Министерства специалисты центра готовы оказать консультации по таким программам и помощь в их приобретении.

**3. Обучение населения**

Центр и музей организовали обучение пользователей домашних компьютеров по собственным учебным программам. Принимаются как дети, так и их родители. Проводится определение карты профессиональных интересов ребенка.

Совместно с физическим факультетом МГУ в его учебных классах организован Воскресный компьютерный университет, где проводится обучение навыкам работы на IBM-совместимых компьютерах, а также подготовлены лекции по обучающим и развива-

ющим программам для семейного использования. Компьютерный университет используется и как средство популяризации физического факультета мгу и привлечения абитуриентов.

#### 4. Интеллектуальные праздники

Разработана новая форма работы с населением — так называемые интеллектуальные праздники (шоу), проводимые в стенах Политехнического музея в дни школьных каникул. В то время, как дети участвуют в компьютерных соревнованиях, родители знакомятся с различными аспектами использования компьютеров в быту, присутствуют на презентациях компьютерных программ, знакомятся с литературой.

Конкретным примером такого интеллектуального шоу является «Шахматная неделя в Политехническом», в рамках которой проводился массовый (около 300 участников) турнир по игре в шахматы с компьютерами, начиная от простейших моделей и кончая сверхсложными, так называемыми «компьютерами Каспарова». В награждении финалистов соревнований приняли участие работники Министерства образования.

При проведении Дня знаний в 1995 г. в рамках подобного интеллектуального шоу проводились массовые (использовалось свыше 30 домашних компьютеров) компьютерные игры и презентации интеллектуально-игровых, обучающих и развивающих программ для детей.

#### 5. Методическая и просветительская работа

Сотрудники центра анализируют вопросы посетителей выставки и координируют свою работу в соответствии с их запросами.

Предпринимаются шаги по работе со средствами массовой информации с целью пропаганды информационной культуры семейного образования. В качестве примера укажем передачу по «Радио России» в разделе «Домашняя академия».

Кроме того, сами интеллектуальные праздники привлекают интерес со стороны СМИ. Так, «Шахматная неделя» освещалась в московских газетах, в некоторых компьютерных изданиях, а также в специализированном журнале «64 — шахматное обозрение».

Накопленный опыт работы по пропаганде компьютерной грамотности обобщен в книге «Нужен ли компьютер вашему ребенку и если нужен, то зачем», где даются необходимые сведения для родителей и их детей по устройству компьютера, аспектам его применения, роли игр в развитии ребенка и т. д. Книга написана в форме диалогов, изобилует простыми и наглядными образами и, как показал опыт, легко читается как детьми, так и взрослыми. Это — первое методическое пособие по информационной культуре семейного образования в отечественной литературе, адресованное непосредственно родителям. (Книга полезна также учителям информатики.)

#### 6. Выводы

Таким образом, благодаря помощи и активной позиции Политехнического музея, при методической поддержке Министерства образования создан прообраз центра пропаганды информационной культуры семейного образования.

Центр «Компьютеры для населения» накопил опыт в деле пропаганды и внедрения интеллектуальной информационной культуры семьи и готов оказать методическую помощь всем заинтересованным организациям и сотрудникам системы Министерства образования, вплоть до конкретных действий по налаживанию системы коммерческого распространения домашних компьютеров и других средств и технологий семейного образования через учреждения Министерства образования.

Министерство рекомендует указанный опыт к широкому применению. Более подробную информацию можно получить в Министерстве образования (Голосов А. И., тел.: (095) 925-20-90) или в Политехническом музее (директор выставки — Ястребов Л. И., тел.: (095) 928-21-38).

# 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 3. Работаем с файлами

Цель данного урока: научиться выполнять основные операции с файлами, работая в среде **Диспетчера файлов**.

#### Основные понятия

**Диспетчер файлов (ДФ)** — это системное средство Windows, позволяющее выполнять все виды работ с файлами: просмотр файловой структуры, копирование, удаление, переименование файлов и каталогов, подготовку дисков к работе, а также запуск программ.

В первой статье из данной серии [1] рассказывалось о том, что система Windows использует при своей работе файловую систему MS DOS. По этой причине, работая в среде Windows, пользователь располагает теми же возможностями для работы с файлами, что и в DOS. Отличается лишь способ манипулирования с файлами: вместо ввода команд на языке DOS или работы с меню оболочки Norton Commander, пользователь выполняет характерные манипуляции с графическими объектами в среде Диспетчера файлов Windows.

Следовательно, пользователь Windows должен быть знаком с основными понятиями файловой системы MS DOS. Речь идет о таких понятиях, как накопитель (магнитный диск), каталог, файл, имя файла, дерево каталогов, путь, текущий диск, текущий каталог.

Если ваши ученики знакомы с этими понятиями, то им достаточно рассмотреть способы изображения основных объектов файловой системы в Windows, после чего можно перейти к освоению работы с Диспетчером файлов. В противном случае требуется основательное объяснение основных понятий файловой системы.

В многочисленной литературе с описаниями MS DOS можно найти достаточно подробную информацию по данным вопросам. Здесь мы лишь приведем основные определения и обратим внимание читателя на методические аналогии, помогающие пониманию материала.

Долговременное хранение информации на компьютере производится на устройствах внешней памяти. На персональных компьютерах основной вид внешней памяти — магнитный диск. Устройство, которое производит запись и чтение информации с магнитного диска, называется дисководом или накопителем на магнитном диске. В дальнейшем его будем кратко называть **накопителем**.

\* Продолжение. Начало см.: Информатика и образование. 1995. № 4, 5.

Компьютер может иметь в своем составе несколько дисководов, работающих с гибкими (floppy) и жесткими (hard) дисками. Последние еще называют винчестерскими дисками. Гибкие диски — сменные устройства, т. е. их можно менять в дисководе; винчестерский диск жестко вмонтирован в компьютер, и пользователь не может его сменить. Как правило, информационная емкость жесткого диска много больше, чем у гибких дисков, к тому же, работа с жестким диском происходит быстрее.

Накопители принято обозначать латинскими буквами с двоеточием. Например, А:, В:, С: и т. д. (нет разницы между использованием больших или маленьких букв). Чаще всего имена А: и В: относятся к гибким дискам, а С: и далее — к жесткому диску.

Из-за большой емкости винчестерского диска его удобно делить на несколько частей, имеющих разные имена: С:, D:, E: и далее. Все они находятся на одном физическом устройстве, однако с точки зрения пользователя выступают как отдельные накопители. Такие части физического диска называют логическими дисками.

**Файловая система** — это определенный способ организации информации на внешних носителях и совокупность средств операционной системы, позволяющих работать с этой информацией.

Центральным понятием и основным объектом файловой системы является файл. **Файл** — наименьший управляемый элемент файловой системы. Файл имеет имя и содержит в себе какую-то однотипную информацию, например текст документа, программу, числовой массив и пр.

В качестве аналогии будем проводить параллель с традиционным «бумажным» способом хранения информации. В таком варианте файл представляется как некоторый документ (текст, чертеж и пр.) на бумажных листах, который определенным образом озаглавлен.

Следующий по величине логический элемент файловой структуры называется каталогом. Продолжая «бумажную» аналогию, каталог будем представлять как папку, в которую можно вложить множество документов, т. е. файлов. Каталог также получает собственное имя (представьте, что на папке написано название).

Разрабатывая интерфейс Windows, авторы исходили именно из такой «бумажной» аналогии. Пиктограмма, изображающая каталог, имеет форму папки:



Каталог может сам входить в состав другого, внешнего по отношению к нему, каталога. Это аналогично тому, как папка вкладывается в другую папку большего размера. Таким образом, каждый каталог может содержать внутри себя множество файлов и вложенных каталогов (говорят — подкаталогов). Каталог самого верхнего уровня, который не вложен ни в какие другие, называется корневым каталогом.

А теперь полную картину файловой структуры представьте себе так: вся внешняя память — это шкаф с множеством выдвижных ящиков. Каждый ящик — аналог накопителя; в ящике — большая папка (корневой каталог); в каждой папке — множество папок и документов (подкаталогов и файлов); самые глубоко вложенные папки хранят в себе только документы (файлы) или могут быть пустыми.

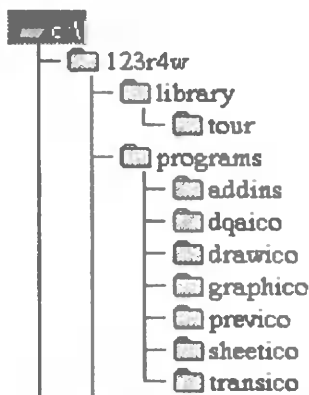
Изображение шкафа с ящиками используется в качестве пиктограммы **Диспетчера файлов**:



Описанную многоуровневую иерархию каталогов называют **деревом каталогов**. Правда, если корневой каталог отнести к самому верхнему уровню этой иерархии, то дерево получается перевернутым, т. е. обращенным корнем вверх. Листьями (конечными элементами этого дерева) являются пустые каталоги или каталоги, содержащие в себе только файлы.

В среде **Диспетчера файлов** дерево каталогов изображается в наглядной форме:





Допустим, что вам нужно найти определенный документ. Для этого надо знать ящик, в котором он находится, а также всю последовательность папок, которые нужно открыть, чтобы добраться до искомого бумаж. Такую последовательность поиска назовем путем к документу. В терминах файловой системы путь к файлу — это последовательность, состоящая из имени накопителя и имен каталогов, начиная от корневого и заканчивая тем, в котором непосредственно хранится файл.

Еще одно понятие — текущий каталог, т. е. это тот каталог, к содержимому которого в данный момент можно обращаться без продвижения вверх или вниз по дереву каталогов. В бумажной аналогии — это последняя раскрытая папка; содержимое которой мы можем обозревать в данный момент. Пиктограмма текущего каталога представляет собой изображение раскрытой папки:



А теперь поговорим о правилах именования каталогов и файлов. Имя каталога может содержать от одного до восьми символов (латинских букв, цифр, а также некоторых других допустимых символов; в имени нельзя употреблять пробел): SYSTEM, WINDOWS, 1CATAL, 2CATAL, PIC\_55 и т. п.

Имя файла состоит из двух частей: собственно имени и расширения, между которыми ставится точка:

<имя>.<расширение>.

Имя — последовательность от 1 до 8 символов (те же правила, как и для каталога). Расширение — необязательная часть, которая может содержать от 1 до 3 символов. Расширение указывает на тип информации, хранящейся в файле. Вот примеры имен файлов: write.exe, letter.wri, format.com, les.28, 5sip#.dat, result (последнее имя не содержит расширения).

Для обозначения группы файлов, относящихся к одному каталогу, используются символы-заполнители: «\*» и «?». Символ «\*» заменяет имя и/или расширение файла. Символ «?» заменяет один знак (см. табл. 1).

Таблица 1

| Пример использования символа-заполнителя | Что обозначает  |
|--|---|
| *.*                                      | все файлы текущего каталога   |
| .bmp                                     | все файлы с расширением .bmp  |
| *a*.wri                                  | все файлы, имя которых начинается со знака «a» и имеет расширение .wri                        |
| a??d.*                                   | все файлы с четырехсимвольными именами с первой буквой «a», последней «d» и любым расширением |

Все файлы можно разделить на программные файлы и файлы данных (докумен-

тов). Программы содержатся в файлах с расширениями EXE, COM, BAT. При активизации этих файлов программы загружаются в оперативную память и выполняются. Файлы других типов содержат разнообразные данные.

Некоторые прикладные программы устанавливают в именах созданных ими файлов стандартные расширения. По этим расширениям можно определить «происхождение» файла данных. Например, расширение .WRI имеют файлы, созданные с помощью редактора Write; расширение .BMP — файлы графического редактора Paintbrush и пр.

И наконец, последнее понятие: полное имя файла — это последовательность, состоящая из пути к файлу и имени самого файла. Элементы, составляющие полное имя, отделяются друг от друга знаком «\» — обратный слэш. Примеры полных имен файлов:

C:\WINDOWS\WRITE.EXE, B:\MYINFORM\MAIL\letter1.wri.

#### Упражнение 1. Управление обстановкой в окне Диспетчера файлов.

В рабочей области окна **Диспетчера файлов** размещаются окна каталогов. В общем случае их может быть несколько. Окно ДФ на рис. 1 содержит одно окно каталогов.

Рабочая область окна каталогов состоит из трех частей. В верхней части располагается список доступных накопителей. Текущий диск выделен маленькой рамкой.

Пространство под списком накопителей состоит из двух панелей. Слева изображается структура дерева каталогов текущего накопителя — панель дерева. Справа располагается список файлов и подкаталогов текущего каталога — панель списка.

#### Задание 1. Запустить Диспетчер файлов:

- > запустить Windows [1].

На экране появится окно ДП с расположенными в нем символами групповых окон. Символ ДФ находится в группе **Главная**.

- > открыть групповое окно **Главная**;
- > дважды щелкнуть левой клавишей мыши на символе **Диспетчер файлов**.

Перед вами откроется окно ДФ (рис. 1).

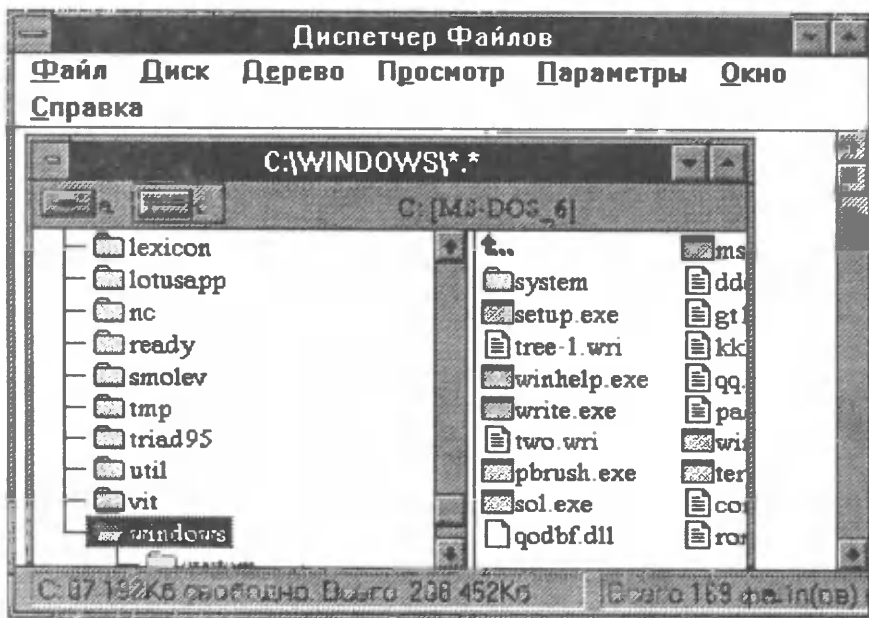


Рис. 1. Окно ДФ

На рис. 1 показан типичный, но не единственный вариант окна ДФ. Если после запуска ДФ вид окна отличается от приведенного на рис. 1, то:

- > выполнить команду **Дерево и каталоги меню Просмотр**.

**Задание 2.** Задать способ отображения панелей окна каталогов:

- > выполнить последовательно команды **Только дерево**, **Только каталоги**, **Дерево и каталоги меню Просмотр**.

Проследить, как изменяется при этом содержимое окна каталогов.

Все установки, сделанные с помощью команд меню **Просмотр**, теряются при выходе из ДФ.

**Задание 3.** Сохранить установки, сделанные с помощью команд меню **Просмотр**:

- > выполнить команду **Сохранять параметры при выходе меню Параметры**.

**Задание 4.** Перераспределить площадь между панелями:

- > установить курсор мыши точно на линию, разделяющую панели (курсор примет вид двунаправленной стрелки, а разделительная линия станет шире);
- > нажать левую клавишу мыши и, не отпуская ее, перемещать курсор;
- > отпустить клавишу мыши после того, как разделительная линия займет желаемое положение.

**Задание 5.** Открыть новое (еще одно) окно каталогов:

*1-й вариант*

- > дважды щелкнуть левой клавишей мыши на пиктограмме накопителя, содержимое которого необходимо увидеть в новом окне;

*2-й вариант*

- > выполнить команду **Новое окно меню Окно**.

Появится новое окно каталогов, содержащее ту же информацию, что и предыдущее активное окно.

Если у вас есть дискета с информацией, то:

- > вставить дискету в накопитель;
- > в новом окне каталогов щелкнуть левой клавишей мыши на пиктограмме этого накопителя.

Содержимое нового окна каталогов изменилось. Теперь в нем располагается дерево и список файлов дискеты.

**Задание 6.** Манипулирование окнами каталогов.

Используя приемы, изложенные на уроке 1, проделать следующее:

- > увеличить размеры окна каталогов до размеров экрана;
- > преобразовать окно каталогов в пиктограмму (пиктограмма окна каталогов располагается в нижней части рабочей области окна ДФ);
- > расположить окна каталогов каскадно (команда **Каскад меню Окно**);
- > расположить окна каталогов без перекрытия рядом друг с другом (команда **Мозаика меню Окно**);
- > закрыть окно каталогов.

**Задание 7.** Активизировать окно каталогов:

- > щелкнуть левой клавишей мыши в любом месте окна.
- Теперь можно работать в этом окне.

**Упражнение 2.** Управление панелью дерева.

Если вы впервые запускаете ДФ, то на левой панели отображается только первый уровень каталогов текущего дискового устройства. Однако пользователь может управлять степенью подробности представления информации о дереве каталогов.

**Задание 1.** Показать один уровень подкаталогов заданного каталога:

*1-й вариант*

- > дважды щелкнуть левой клавишей мыши на имени или пиктограмме каталога, ветви которого вы хотите показать;

*2-й вариант*

- > щелкнуть левой клавишей мыши на имени или пиктограмме каталога, ветви которого вы хотите показать;
- > выполнить команду **Показать один уровень меню Дерево**.

**Задание 2.** Показать все уровни подкаталогов заданного каталога:

- > щелкнуть левой клавишей мыши на имени или пиктограмме каталога, ветви которого вы хотите показать;
- > выполнить команду **Показать ветви** меню **Дерево**.

**Задание 3.** Показать все ветви дерева каталогов:

- > выполнить команду **Показать все** меню **Дерево**.

**Задание 4.** Скрыть ветви каталога:

*1-й вариант*

- > дважды щелкнуть левой клавишей мыши на имени или пиктограмме каталога, ветви которого вы хотите скрыть;

*2-й вариант*

- > щелкнуть левой клавишей мыши на имени или пиктограмме каталога, ветви которого вы хотите скрыть;
- > выполнить команду **Спрятать ветви** меню **Дерево**.

**Задание 5.** Дополнить дерево каталогов информацией о подкаталогах:

- > выполнить команду **Отмечать расширяемые ветви** меню **Дерево**.

Теперь каждый каталог, имеющий подкаталоги, помечен знаком «+». Знак «+» означает, что каталог имеет скрытые в данный момент ветви и вы можете их показать. Знак «-» говорит о том, что показаны все подкаталоги.

**Задание 6.** Отменить действие команды **Отмечать расширяемые ветви**:

- > выполнить эту команду еще раз.

### Упражнение 3. Управление панелью списка

Панель списка содержит информацию о файлах и подкаталогах текущего каталога. Пиктограммы здесь более разнообразны, чем на панели дерева. Файлы с программами изображаются следующим символом:



Файлы данных (документов) имеют вид листа с загнутым углом. Система различает **связанные** и **свободные** документы.

Связанные документы отличаются важным свойством: системе WINDOWS известны приложения для их обработки. Связь между приложениями группы РЕКВИЗИТЫ и соответствующими им документами устанавливается автоматически. Расширение документа говорит о том, каким приложением этот документ подготовлен (табл. 2).

Таблица 2

| Расширение документа | Каким приложением документ создан |
|----------------------|-----------------------------------|
| .BMP                 | Paintbrush                        |
| .CAL                 | Календарь                         |
| .CRD                 | Картотека                         |
| .TXT                 | Блокнот                           |
| .WRI                 | Write                             |

Отличаются пиктограммы, обозначающие связанные и свободные документы:



свободный документ



связанный документ

Основным средством управления видом панели списка являются команды меню **Просмотр**. С помощью этих команд можно изменять полноту информации о файлах;

«фильтровать» список файлов, т. е. оставлять в списке только определенную группу файлов; упорядочивать список по определенным признакам.

**З а д а н и е 1.** Установить текущий каталог:

- > щелкнуть левой клавишей мыши на имени или пиктограмме нужного каталога на дереве каталогов.

Теперь на правой панели — список файлов и подкаталогов выбранного вами каталога.

**З а д а н и е 2.** Задать отображение всех характеристик файлов и каталогов:

- > выполнить команду **Все характеристики файлов** меню **Просмотр**.

**З а д а н и е 3.** Задать отображение только имен файлов и каталогов:

- > выполнить команду **Имя** меню **Просмотр**.

**З а д а н и е 4.** Задать отображение отдельных характеристик файлов и каталогов:

- > выполнить команду **Отдельные характеристики** меню **Просмотр**

Появится диалоговое окно (см. рис. 2);

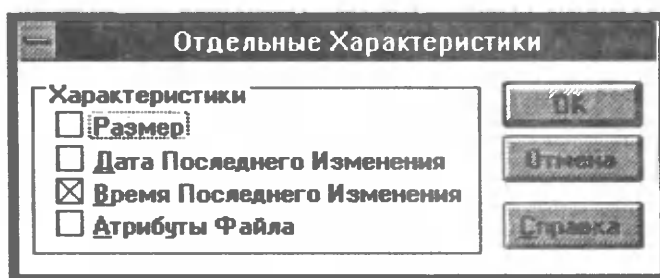


Рис. 2. Диалоговое окно для выбора отображаемых характеристик файлов

- > выберите характеристики, которые вы хотите увидеть на правой панели.

**З а д а н и е 5.** Отобразить в списке файлов только информацию о документах:

- > выполнить команду **По типу файла** меню **Просмотр**.

Откроется диалоговое окно (см. рис. 3);

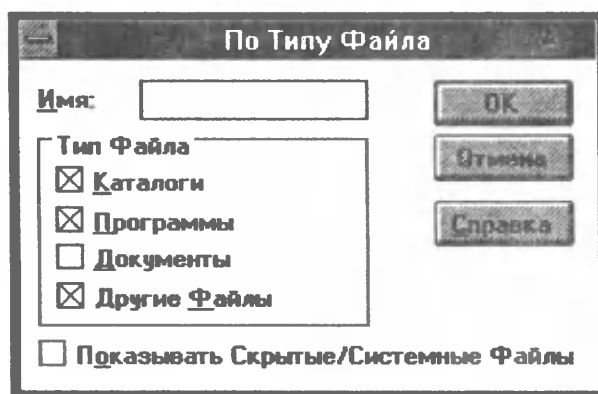


Рис. 3. Диалоговое окно для выбора типов отображаемых файлов

- > включить квадратное поле выбора **Документы**;
- > выключить остальные квадратные поля выбора.

Фильтр можно установить так, чтобы на правой панели отображалась информация только о файлах определенного типа.

**Задание 6.** Отобразить в списке файлов информацию только о файлах текстового редактора:

- > выполнить команду **По типу файла** меню **Просмотр**;
- > в диалоговом окне (см. задание 5) в поле **Имя** ввести текст \*.WRI.

**Задание 7.** Отсортировать файлы на правой панели:

- > выполнить (последовательно) команды **Сортировка по имени**, **Сортировка по типу**, **Сортировка по размеру**, **Сортировка по дате** меню **Просмотр**.
- Проследить за изменением содержимого правой панели.

**Упражнение 4. Работа с файлами и каталогами.**

Научившись управлять видом панелей окна каталогов, перейдем к освоению приемов работы с файлами и каталогами. **Диспетчер файлов** позволяет выполнять следующие действия:

- создавать и уничтожать каталоги;
- перемещать файлы из одного каталога в другой;
- копировать файлы из одного каталога в другой;
- переименовывать файлы и каталоги;
- уничтожать файлы;
- искать файлы на носителе по имени;
- запускать приложения.

При копировании файлы целиком переписываются из исходного каталога в другой каталог. При перемещении производится копирование файлов и, кроме того, стирание их в исходном каталоге. Если необходимо переместить (скопировать) файлы в пределах одного накопителя, то потребуется только одно окно каталогов. Если нужно переместить (скопировать) файлы на другой диск, желательно организовать два окна каталогов. **Важно помнить, что ПЕРЕМЕЩАТЬ МОЖНО ТОЛЬКО ТЕ ФАЙЛЫ И КАТАЛОГИ, КОТОРЫЕ ВЫ СОЗДАЛИ САМИ.**

Файлы или каталоги, с которыми должна быть произведена какая-то работа, предварительно выделяются. Выделенные объекты отмечаются закрашенной прямоугольной рамкой.

**Задание 1.** Выделить файл (каталог):

- > щелкнуть левой клавишей мыши на имени файла (каталога).

**Задание 2.** Выделить несколько следующих друг за другом файлов (каталогов):

- > щелкнуть левой клавишей мыши на первом файле (каталоге);
- > установить курсор мыши на последний файл (каталог); при необходимости воспользуйтесь линейкой прокрутки;
- > нажать клавишу **Shift** и, не отпуская ее, щелкнуть левой клавишей мыши на имени последнего файла (каталога).

Если вы выделили слишком много (мало) файлов (каталогов), можно скорректировать полученный список.

**Задание 3.** Увеличить (уменьшить) список выделенных файлов (каталогов):

- > установить курсор мыши на последний файл (каталог), который необходимо выделить
- > нажать клавишу **Shift** и, не отпуская ее, щелкнуть левой клавишей мыши.

**Задание 4.** Выделить несколько файлов (каталогов), следующих не по порядку:

- > нажать и не отпускать клавишу **Ctrl**;
- > последовательно щелкнуть левой клавишей мыши на именах файлов (каталогов), которые необходимо выделить;
- > отпустить клавишу **Ctrl**.

Если по ошибке вы выделили слишком много файлов (каталогов), то некорректные выделения можно отменить.

**Задание 5.** Отменить выделение файла (каталога):

- > нажать и не отпускать клавишу **Ctrl**;
- > щелкнуть левой клавишей мыши на имени выделенного файла (каталога);
- > отпустить клавишу **Ctrl**.

**Задание 6. Создать каталог:**

- > выделить каталог, в который вы хотите добавить подкаталог;
- > выполнить команду **Создать каталог** меню **Файл**.  
На экране появится диалоговое окно (см. рис. 4);

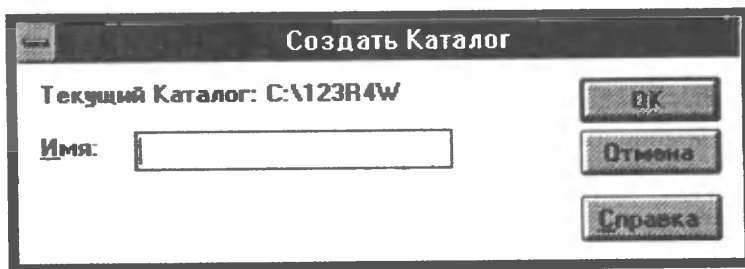


Рис. 4. Диалоговое окно для создания каталога

- > ввести имя создаваемого каталога (например, MY\_DIR) в поле **Имя** диалогового окна.

**Задание 7. Переместить файлы (каталоги):**

- > выделить файлы (каталоги), которые нужно переместить;
- > нажать и не отпускать клавишу Shift;
- > подвести курсор мыши к выделенным файлам;
- > нажать левую клавишу мыши и, не отпуская ее, перетащить выделенные файлы в результирующий каталог;
- > отпустить клавишу мыши;
- > отпустить клавишу Shift.

*Замечание.* Если вы перемещаете файлы или каталоги на одном и том же дисковом устройстве, то можно не использовать клавишу Shift.

**Задание 8. Скопировать файлы (каталоги):**

- > выделить файлы (каталоги), которые вы хотите скопировать;
- > нажать и не отпускать клавишу Ctrl;
- > подвести курсор мыши к выделенным файлам;
- > нажать левую клавишу мыши и, не отпуская ее, перетащить выделенные файлы в результирующий каталог;
- > отпустить клавишу мыши;
- > отпустить клавишу Ctrl.

*Замечание.* Если вы копируете файлы или каталоги на другое устройство, то можно не использовать клавишу Ctrl.

**Задание 9. Переименовать файл (каталог):**

- > выделить файл (каталог), который вы хотите переименовать;
- > выполнить команду **Переименовать** меню **Файл**.

Появится диалоговое окно:

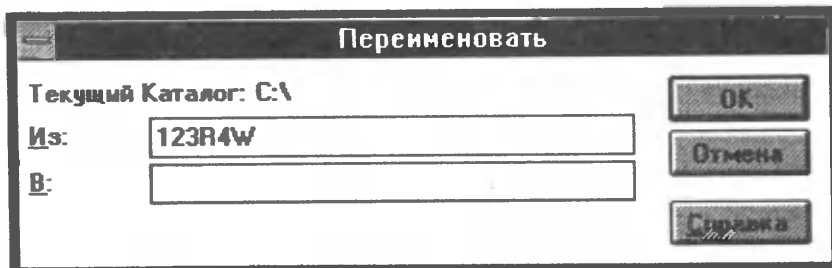


Рис. 5. Диалоговое окно для переименования файлов и каталогов

В поле **Из Диспетчер файлов** записывает имя файла (каталога), который должен быть переименован:

- > ввести новое имя в поле ввода В.

**Задание 10.** Удалить файлы (каталоги):

- > выделить удаляемые файлы (каталоги);
- > нажать клавишу Delete или выполнить команду **Удалить** меню **Файл**.

Нужно быть очень осторожным при выполнении команды удаления файла. Далеко не в каждом случае существует возможность восстановить утерянную информацию. Поэтому желательно перед удалением очередного файла увидеть предупреждающее сообщение, в ответ на которое пользователь должен категорически подтвердить необходимость команды удаления.

Если вы не увидели предупреждающего сообщения при выполнении задания 10, рекомендуем задание 11.

**Задание 11.** Ввести предупреждающие сообщения при удалении файлов:

- > выполнить команду **Подтверждение** меню **Параметры**.
- Откроется диалоговое окно (см. рис. 6);

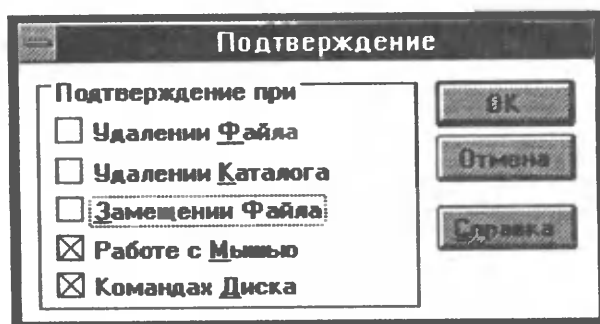


Рис. 6. Диалоговое окно для формирования подтверждающих сообщений

- > включить квадратные поля выбора **Удалении файла** и **Удалении каталога**.

**Задание 12.** Найти файл:

- > выполнить команду **Поиск** меню **Файл**.
- Откроется диалоговое окно:

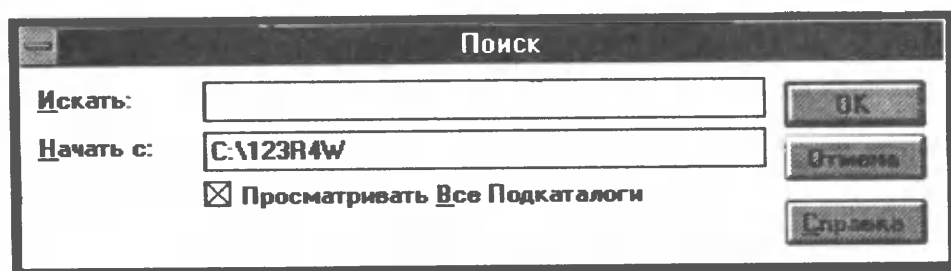


Рис. 7. Диалоговое окно поиска файлов

В поле **Начать с** записывается текущий каталог, который выделен в левой панели. Если выключено квадратное поле выбора **Просматривать все подкаталоги**, поиск производится только в текущем каталоге, в противном случае просматриваются текущий каталог и все его подкаталоги;

- > ввести в поле **Искать** имя искомого файла.

В поле **Искать** для поиска группы файлов можно использовать символы-заполнители «\*», «?».



Если ДФ найдет один или несколько файлов, то раскроется еще одно окно, в котором будут перечислены все найденные файлы. В противном случае ДФ выдаст сообщение об отсутствии искомым файлов.

#### Упражнение 5. Использование ДФ для запуска программ.

Непосредственно из ДФ можно запустить на исполнение программы, хранящиеся в файлах с расширениями EXE, COM.

Документы не могут быть запущены на выполнение и всегда нуждаются в программе, которая бы их создавала и обрабатывала. Однако при инициализации связанных документов (см. выше) происходит запуск соответствующего приложения и загрузка документа.

**Задание 1.** Запустить приложение (например, notepad.exe) на выполнение:

- > дважды щелкнуть левой клавишей мыши на пиктограмме или имени приложения.

**Задание 2.** Запустить приложение для обработки связанного документа:

- > дважды щелкнуть левой клавишей мыши на имени или пиктограмме документа. Таким образом, вы запустили приложение и одновременно загрузили документ.

#### Упражнение 6. Форматирование дискетты.

Форматирование дискетты аналогично разлиновке в клеточку чистого листа бумаги, после чего при записи на лист в каждую клетку будет помещаться один символ. Для дисков такая разлиновка называется форматированием. Она синхронизирует работу магнитного устройства чтения-записи, определяя позиции на поверхности диска, в которые будет записываться и откуда будет читаться двоичная информация. ЕСЛИ ДИСК УЖЕ СОДЕРЖАЛ КАКУЮ-ТО ИНФОРМАЦИЮ, ТО В РЕЗУЛЬТАТЕ ФОРМАТИРОВАНИЯ ОНА БУДЕТ УНИЧТОЖЕНА.

**Задание 1.** Отформатировать дискетту:

- > вставить дискетту в накопитель;
- > выполнить команду **Форматирование** диска меню **Диск**. Откроется диалоговое окно (см. рис. 8);

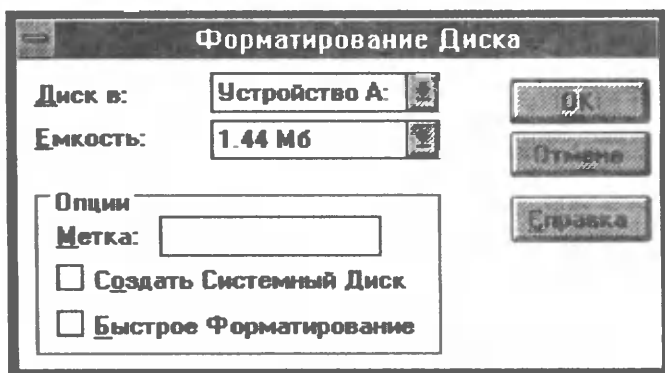


Рис. 8. Диалоговое окно для форматирования дискет

- > указать дисковод в поле **Диск в:**
- > указать объем, который должна иметь дискета в поле **Емкость**. При форматировании можно создать системную дискету. Для этого
- > включить квадратное поле выбора **Создать системный диск**. С системной дискеты можно загрузить компьютер.

#### Литература

1. Залогова Л. А., Семакин И. Г. Уроки по Windows//Информатика и образование. 1995. № 4, 5.
2. Фаненштих К., Хаселир Р. Операционная среда Windows. М.: Изд. ЭКОМ, 1994.

Л. А. Залогова, И. Г. Семакин,

г. Пермь

## БАЗОВЫЙ КУРС ОИВТ: «ПЕРМСКАЯ ВЕРСИЯ» (часть 8)\*

### Тема 11. Моделирование знаний на ЭВМ

Данной статьей начинаются публикации методических материалов, относящихся ко второй части базового курса ОИВТ, ориентированной на IX класс. С начала публикаций прошло значительное время, в течение которого произошло переосмысление авторами некоторых моментов курса. Основное изменение коснулось тематического плана для IX класса, который был опубликован в ИНФО, 1994, № 5. Тема «Моделирование знаний на ЭВМ» переместилась с номера 15 на номер 11. Последовательность остальных тем не изменилась.

#### От баз данных к базам знаний

В теме «Компьютер дает справку» отмечалось, что информация, относящаяся к некоторому объекту и хранящаяся в базе данных, представляет собой и н ф о р м а - ц и о н н у ю модель этого объекта. Расписание движения поездов, самолетов, таблица Менделеева, генеалогическое дерево царской фамилии и пр. — все это примеры баз данных (БД). База данных — это некоторым образом структурированная совокупность ф а к т о в, относящихся к определенному объекту.

А можно ли, допустим, учебник математики назвать базой данных по математике или учебник физики — базой данных по физике? Очевидно, нет! В учебниках содержатся не только факты, но и правила. Например:

факты: « $2 \times 2 = 4$ »; «отношение длины окружности к диаметру равно 3,14159...», «атом состоит из ядра и электронов»;

правила: «если треугольник — прямоугольный, то для него выполняется теорема Пифагора»; «число называется простым, если оно делится без остатка только на единицу и само на себя»; «однородные члены предложения разделяются запятыми» и пр.

Таким образом, учебник уместнее назвать базой знаний в определенной области знаний.

Совокупность фактов и правил, хранящихся в памяти ЭВМ, будем называть компьютерной базой знаний.

Но достаточно ли заучить все учебники математики, для того чтобы стать выдающимся математиком?

Представим себе такую ситуацию: два ученика, Саша и Миша, подготовились к экзамену по математике. Саша, обладая феноменальной памятью, выучил наизусть весь учебник и «зазубрил» все решения задач. Убеденный в надежности своей памяти, он был уверен в успешном результате экзамена. Его товарищ, не имея такой выдающейся памяти, избрал другой метод подготовки к экзамену: постарался все понять, разобрался в логике предмета, сам прорешал все задачи.

Результат оказался неожиданным для друзей: Саша «завалил» экзамен, а Миша получил пятерку.

Разбирая с учениками эту ситуацию, поставьте перед ними вопрос: в чем разница

В полном объеме «Базовый курс ОИВТ: пермская версия» опубликован в книге: Залогова Л. А., Русакова С. В., Семакин И. Г., Хеннер Е. К., Шестакова Л. В. Основы информатики и вычислительной техники в базовой школе: Пособие для учителя/Под ред. И. Г. Семакина. Пермь, 1995. Справки по адресу: 614600, г. Пермь, ГСП, бул. Гагарина, д. 37-а, центр «Кадриформ». Тел.: (3422) 66-91-90, 66-09-96.

\* Продолжение. Начало см.: Информатика и образование. 1994. № 5, 6; 1995. № 1, 2, 3, 5.

между знаниями Саши и Миши? Почему это привело к таким разным результатам на экзамене?

Знания Саши — это зазубренный набор слов и формул, которые он мог только повторить. Миша же, даже в тех ситуациях, когда забывал какие-то правила или формулы, сам выводил их.

Его успех объясняется тем, что он знал основные понятия (факты) и основные правила предмета и обладал логикой мышления, с помощью которой получал другие факты и правила, следующие из основных.

Саша же, не владея логикой, показал полную неспособность мыслить в рамках этого предмета и поэтому получил двойку.

Из этого примера следует вывод о том, что творческое мышление человека основано на трех составляющих: знании фактов, знании правил и логики.

О том, что знания человека, т. е. информация, которой он обладает, состоит из фактов и правил, говорилось в теме «Человек и информация». Следует напомнить ученикам и мысль о том, что компьютер по своей организации моделирует информационную функцию человека. Компьютер, в памяти которого хранится база данных, относящаяся к какой-то области знаний, подобен человеку, знающему только факты. Такой человек способен ответить лишь на тот вопрос, ответ на который он помнит. Заучив, например, таблицу умножения, ученик автоматически отвечает на вопрос: «Сколько будет  $5 \times 5$ ?» и т. п. Так и информационно-поисковая система (ИПС) на ЭВМ может отвечать только на те вопросы, ответы на которые непосредственно хранятся в ее базе данных, сделав соответствующую выборку. Чем более «умной» мы хотим сделать информационно-поисковую систему, тем больший размер базы данных для этого требуется.

Однако для повышения интеллекта компьютера можно выбрать и другой путь. Для этого нужно иметь средства построения компьютерных баз знаний и моделирования на ЭВМ логики рассуждений человека.

Рассмотрим традиционный для этой темы пример информации о родственных отношениях между членами семьи. Сначала речь пойдет о семье из трех человек, отношения между которыми выражены схемой:



Если мы хотим создать реляционную базу данных, из которой можно узнать, кто чья мама или дочь, то ее следует организовать так:

Таблица 1

| БД «МАМА»  |          |
|------------|----------|
| имя_дочери | имя_мамы |
| Ольга      | Нина     |
| Тяня       | Ольга    |
| Лена       | Ольга    |

Напомним, что в терминологии реляционных баз данных такая таблица называется **отношением**, и ее структура описывается:

## МАМА (Имя дочери, имя\_ мамы).

Поле «Имя дочери» подчеркнуто, так как оно является ключевым. Для того чтобы узнать, кто мама Ольги, запрос можно построить так (в терминологии dBASE):

**.LIST имя\_ мамы, FOR имя\_дочери="Ольга".**

А для того чтобы получить список имен дочерей Ольги, нужно отдать команду:

**.LIST имя\_дочери, FOR имя\_ мамы="Ольга".**

Из этой базы данных нельзя получить информацию о бабушках и внуках, хотя, очевидно (нам с вами, но не ИПС), что Нина является бабушкой Тани и Лены. Чтобы такую информацию можно было получить, в базу данных нужно включить еще одну таблицу (отношение) «БАБУШКА»:

Таблица 2

| БД «БАБУШКА» |             |
|--------------|-------------|
| имя_внучки   | имя_бабушки |
| Таня         | Нина        |
| Лена         | Нина        |

Теперь, чтобы узнать, кто бабушка Тани, нужно перейти к работе с файлом, содержащим таблицу «БАБУШКА» и отдать команду:

**.LIST имя\_бабушки, FOR имя\_внучки="Таня".**

В том же духе нужно продолжать дальше, если мы хотим получать из базы данных сведения о сестрах. Придется опять завести новую таблицу. Правда, можно пойти и другим путем: не размножать отношения, а увеличивать число полей в одной таблице. Вот другой вариант таблицы со сведениями о тех же родственниках:

Таблица 3

| БД «РОДСТВЕННИКИ» |             |          |            |
|-------------------|-------------|----------|------------|
| персона           | имя_бабушки | имя_мамы | имя_сестры |
| Нина              | не знаю     | не знаю  | не знаю    |
| Ольга             | не знаю     | Нина     | не знаю    |
| Таня              | Нина        | Ольга    | Лена       |
| Лена              | Нина        | Ольга    | Таня       |

Включение в БД сведений о других родственных связях снова приведет к ее разрастанию.

Однако ясно, что такие базы содержат избыточную информацию. Зачем, например, нужно указывать отдельно то, что Нина — бабушка Тани, если это и так следует из того, что Нина — мама Ольги, а Ольга — мама Тани. Если бы каким-то образом можно было сообщить компьютеру правило о том, что «мама мамы есть бабушка», и он обладал бы способностью делать логические выводы на основании известных ему правил и фактов, то для выяснения вопроса о бабушке было бы достаточно первой таблицы. Если же еще сообщить компьютеру, что «две персоны — сестры, если у них общая мама», то из первой таблицы можно установить, что Таня и Лена — сестры.

Еще раз повторим: компьютерная система, хранящая в памяти факты и правила, относящиеся к определенной предметной области, называется базой знаний.

Итак, база знаний о родственных отношениях в рассмотренной нами семье состоит из таблицы 1 (базы данных) и двух правил: «мама мамы есть бабушка» и «две персоны — сестры, если у них общая мама».

Следовательно, база данных входит в базу знаний как компонента.

Знакомясь в теме «Компьютер дает справку» с базами данных, мы узнали, что система программного обеспечения, с помощью которой создаются БД, называется системой управления базами данных (СУБД). С помощью СУБД

организуется поиск нужной информации в базе данных. Примером такой СУБД является dBASE.

Для построения базы знаний и извлечения из нее нужной информации также существует специальное программное обеспечение. Наиболее известной системой такого рода является система Пролог.

Пролог — язык логического программирования (logic programming language), появившийся в 70-х годах во Франции и связанный с работами в области создания искусственного интеллекта. Принято считать Пролог языком пятого поколения ЭВМ [1, 2, 3]. С его помощью разрабатываются сложные интеллектуальные компьютерные системы управления производственными процессами, перевода с одного языка на другой, математических вычислений в аналитической форме, интеллектуального интерфейса с человеком, шахматной игры, экспертизы в различных предметных областях и многое другое.

Высшим уровнем продукции, создаваемой на Прологе, являются э к с п е р т н ы е с и с т е м ы. Разработка экспертных систем, в свою очередь, составляет важнейшее направление раздела информатики, который называется «Искусственный интеллект» [4].

Экспертная система — это компьютерная программа, которая в конкретной предметной области проявляет познания, равнозначные знаниям высококвалифицированного специалиста-эксперта.

Например, это может быть система медицинской диагностики; система определения видов растений, насекомых и других биологических объектов; система помощи конструктору автомобилей, позволяющая ему выбрать наилучшее конструкторское решение, материал, технологию и пр.

Любая экспертная система содержит в себе три основных компоненты :

- базу знаний в соответствующей предметной области;
- механизм вывода (реализованный программными средствами способ логических рассуждений и выводов);
- систему пользовательского интерфейса (т. е. удобную для пользователя среду общения, язык общения, методику общения).

В данном разделе курса информатики основное внимание будет уделено знакомству с базами знаний и приведены краткие сведения о механизме вывода. Вопрос о способах создания пользовательского интерфейса освещаться не будет, так как с этим средством ученики познакомятся, работая с демонстрационной экспертной системой.

Первым опытом изложения элементов Пролога в школьном курсе информатики стал учебник В. А. Каймина, Е. А. Щеголева и др. [5]. В своих дальнейших разработках этот авторский коллектив продолжает развитие линии логического программирования в курсе ОИВТ [6, 7, 8], сделав эту линию ведущей.

В данной статье также излагаются элементы Пролога. Однако делается это в очень узких рамках. Учителю предлагается вообще не употреблять терминологию «Пролог — язык логического программирования». Будем говорить о с и с т е м е П р о л о г, назначение которой — создание базы знаний и получение ответов на поставленные к ней вопросы (цели), и интерпретировать систему Пролог как средство м о д е л и р о в а н и я з н а н и й н а Э В М.

### Основные понятия Пролога

В Прологе базовыми понятиями являются: ф а к т, п р а в и л о, ц е л ь.

#### Факты

Вернемся к таблице 1 и будем ее интерпретировать как совокупность фактов, которые в обычном разговорном языке выражаются предложениями:

Нина — мама Ольги;  
Ольга — мама Тани;  
Ольга — мама Лены.

Эти предложения устанавливают отношения между Ниной и Ольгой, Ольгой и Таней, Ольгой и Леной, которые можно записать в такой форме:

| Объект | Отношение | Объект |
|--------|-----------|--------|
| Нина   | мама      | Ольга  |
| Ольга  | мама      | Таня   |
| Ольга  | мама      | Лена   |

Чтобы перейти к форме записи этих отношений на Прологе, необходимо сделать в них следующую перестановку:

| Отношение | Объект | Объект |
|-----------|--------|--------|
| мама      | Нина   | Ольга  |
| мама      | Ольга  | Таня   |
| мама      | Ольга  | Лена   |

Окончательно запись на Прологе этих трех фактов имеет вид:

мама (нина, ольга);  
мама (ольга, тания);  
мама (ольга, лена).

Термин «факт» в Прологе имеет определенный формальный смысл и указывает на то, что выполнено некоторое отношение между объектами. Общий вид записи факта на Прологе:

<имя факта> (<аргументы>).

Следовательно, в нашем примере факт имеет имя «мама», а «нина», «ольга», «таня», «лена» — аргументы. В соответствии с синтаксисом Пролога имена фактов и аргументов пишутся строчными буквами. В общем случае факт может содержать произвольное количество аргументов (0, 1, 2, ...).

В случае факта с одним аргументом уже нельзя говорить об отношении между объектами, так как объект всего один. Например, фактом является:

отличник (Сергей).

В этом случае факт отражает некоторый признак объекта-аргумента.

Определяя с помощью фактов отношения между объектами, нужно учитывать последовательность перечисления аргументов. Например, факты мама (нина, ольга) и мама (ольга, нина) имеют разный смысл. В нашем примере первый факт имеет место, второй — нет. Первый аргумент отвечает на вопрос «кто (что)?», второй — «по отношению к кому (чему)?».

Вот еще один пример. Известно, что

Мария знает английский.

Этот факт на Прологе запишется так:

знает (мария, английский).

Пример. Записать на Прологе утверждения естественного языка.

Утверждение на естественном языке

Мария — женщина  
Иван владеет пекарней  
Петров владеет автостоянкой  
Петров — собственник  
Иван — собственник  
Сидоров — учитель  
Макаров — учитель  
Джон дает Мэри книгу  
Иван дает Марии альбом  
Мэри нравится Джону  
Джон нравится Мэри

Соответствующий факт на Прологе

женщина (мария)  
владеет (иван, пекарня)  
владеет (петров, автостоянка)  
собственник (петров)  
собственник (иван)  
учитель (сидоров)  
учитель (макаров)  
дает (джон, мэри, книга)  
дает (иван, мария, альбом)  
нравится (мэри, джон)  
нравится (джон, мэри)

## Правила

Рассмотрим два факта:

знает (мария, английский);  
знает (иван, английский).

Смысл их понятен:

«Мария знает английский»

и

«Иван знает английский».

Очевидно, те же сведения несут следующие два предложения:

«Мария знает английский»

и

«Иван знает то же, что и Мария».

Но если первое предложение — факт, то второе уже можно назвать правилом. Оно обладает большей общностью, чем частный факт. Пусть, например, Мария выучила испанский язык. Тогда, если справедливо предыдущее правило, из этого будет следовать, что Иван знает английский и испанский.

В данном примере правило связывает между собой два факта союзом «если». Это правило можно переформулировать так:

«Иван знает X, если Мария знает X».

Под символом X подразумевается любое существительное, обозначающее все, что можно знать (английский, история, информатика и пр.). В терминологии Пролога в данном примере символ X называется переменной.

Переменная не является именем конкретного объекта, она используется для обозначения различных объектов. В Прологе принято соглашение, позволяющее отличать переменные от имен конкретных объектов: имя, начинающееся с прописной буквы, рассматривается как переменная. Для формулировок правил характерно использование переменных.

Запись правила на Прологе состоит из двух частей, разделенных знаком «:-» (если). Рассмотренное выше правило запишется на Прологе в виде:

знает (иван, X) :- знает (мария, X).

Правила содержат условную часть (справа от знака :-) и часть вывода (слева от знака :-). Условную часть называют телом правила, а вывод — головой. Если тело правила — истинно, то следствием из него является голова.

Чем больше аргументов, входящих в правило, являются переменными, тем большей общностью оно обладает. Например, следующее правило:

знает (X, Y) :- знает (мария, Y)

надо понимать так: «все знают то, что знает Мария».

В следующем правиле все аргументы являются переменными:

дочь (A, B) :- мама (B, A).

Расшифруйте сами его смысл!

Факты и правила — предложения, из которых строится база знаний. Следовательно, база знаний «Дочки-матери» должна быть описана так:

**Факты:**           мама (нина, ольга);  
                      мама (ольга, тania);  
                      мама (ольга, лена);

**Правила:**       дочь (A, B) :- мама (B, A)

Базу знаний из второго примера назовем «Знатоки»:

**Факты:**           знает (мария, английский);

**Правила:**       знает (иван, X) :- знает (мария, X)

### Цель

Для чего же создается база знаний? Ответ на этот вопрос очевиден: как и база данных, база знаний создается для того, чтобы можно было получать из нее информацию в форме ответов на различные вопросы.

В терминологии Пролога вопрос к базе знаний называется целью. Может быть сформулировано два вида целей (вопросов):

1) подтвердить справедливость факта; в результате получается ответ «да», если факт содержится в базе знаний или выводится (доказывается) с использованием предложений базы знаний; в противном случае получается ответ «нет».

2) перечислить все значения переменной, указанной в запросе, удовлетворяющие фактам и правилам базы знаний.

Перед вопросом будем ставить знак вопроса (?) согласно синтаксису некоторых версий Пролога, о которых будет сказано ниже.

Вопрос первого типа к БЗ «Знатоки» в разговорной форме:

«Знает ли Иван английский?»

На Прологе этот же вопрос имеет вид:

? знает (иван, английский).

Ответ на него будет: «да».

Вопрос такого типа к БЗ «Дочки-матери»:

? мама (таня, нина)

означает: «Является ли Таня мамой Нины?». Ответ, очевидно, будет: «нет». А на вопрос

? дочь (лена, ольга),

т. е. является ли Лена дочерью Ольги, будет получен положительный ответ. Для ответа на вопрос Пролог проводит логические рассуждения, используя предложения БЗ.

Теперь рассмотрим примеры целей (вопросов) второго типа. Вот как запишется на Прологе вопрос: «Что знает Иван?»:

? знает (иван, X).

Система должна сообщить все значения переменной X, следующие из базы знаний. Ответом будет

английский.

А следующий вопрос:

? мама (ольга, X)

означает: «Кому Ольга приходится мамой?» Ответом будет:

таня  
лена.

### Как Пролог-система отвечает на вопросы

Если по каким-либо причинам учитель вынужден сократить изучаемый материал по данной теме программы (недостаток времени, желание уделить больше внимания другим темам и пр.), то он может упустить объяснение механизма вывода Пролога, ограничившись только поверхностным описанием баз знаний и вопросов к ним.

Однако учитель должен хорошо понимать, что с точки зрения полноты изучения темы обсуждение алгоритма механизма вывода, которое следует ниже, очень актуально. В нем заключается описание реализации в Прологе логики человеческих рассуждений. Именно соединение информации, заложенной в базе знаний с механизмом логических рассуждений, позволяет говорить о моделировании человеческого интеллекта.

Во-первых, следует спросить учеников: «Как вы получаете ответ на вопрос: «Что знает Иван?», исходя из базы знаний «Знатоки?»



Дети ответят приблизительно так: «Из правила о том, что Иван знает то же, что знает Мария, логически следует, что Иван знает английский».

Что значит «логически следует»?

Навыки логического мышления частично присущи человеку генетически, но большей частью вырабатываются в процессе жизнедеятельности.

Разгадка сущности человеческой логики — одна из древнейших научных проблем. Первый удачный опыт формального описания логики принадлежит древнегреческому философу Аристотелю (более двух тысяч лет назад).

До XIX века формальная логика Аристотеля считалась непререкаемой. В середине XIX века в эту науку существенный вклад внес английский математик де Морган (правила Моргана). Математический аппарат логики развил современник и соотечественник Моргана — Дж. Буль (булева алгебра). Формальная логика активно развивается и в XX столетии [3, 5].

Пролог-система моделирует человеческую логику, опираясь на математический аппарат формальной логики, через алгоритмы, которые называют механизмом вывода Пролога.

Рассмотрим алгоритм механизма вывода при поиске ответа на первый тип вопроса (в Прологе это называется доказательством цели).

Еще раз выпишем БЗ «Знатоки».

**Факты:** знает (мария, английский);

**Правила:** знает (иван, X) :- знает (мария, X).

И задана цель:

? знает (иван, английский).

Факты и правила используются для доказательства цели. Пролог ищет факты и головы правил, сопоставимые с целью. Два факта сопоставимы (или соответствуют друг другу), если выполняются следующие три условия:

1. Имена предикатов одинаковы (побуквенное совпадение).
  2. Предикаты имеют равное количество аргументов.
  3. Аргументы, расположенные в одинаковых позициях, сопоставимы.
- Сопоставление аргументов осуществляется по следующим правилам:
- 3.1. имена конкретных объектов сопоставимы, если они совпадают;
  - 3.2. переменная сопоставима с именем конкретного объекта;
  - 3.3. переменная сопоставима с другой переменной.

Пролог просматривает утверждения в том порядке, в каком они вводились в программу (на печатной странице это соответствует просмотру сверху вниз). Поэтому сначала сравнивается цель

? знает (иван, английский)

и факт

знает (мария, английский).

Первые аргументы несопоставимы (см. правило 3.1). Следовательно, попытка сопоставить цель и факт unsuccessful. Пролог продолжает поиск следующего предиката «знает» и находит правило:

знает (иван, X) :- знает (мария, X).

Цель и голова правила сопоставимы, так как выполнены условия 1, 2 и 3.2. В результате сопоставления переменная X означается, т. е. принимает значение «английский». Теперь каждому появлению переменной X в правиле соответствует значение «английский»:

знает (иван, английский) :- знает (мария, английский).

На следующем этапе Пролог должен проверить истинность тела правила для того, чтобы доказать или опровергнуть его голову. Таким образом, подцель

? знает (мария, английский)

создается самим Прологом. Для доказательства подцели Пролог просматривает базу знаний и находит факт

знает (мария, английский),

который сопоставим с подцелью. Следовательно, голова правила действительно имеет место, и доказана истинность цели.

Новый факт успешно выведен, хотя он нигде в программе явно указан не был.

Теперь обсудим механизм вывода ответа на вопросы второго типа. Пусть база содержит факты

|                             |     |
|-----------------------------|-----|
| знает (иван, английский).   | (1) |
| знает (мария, информатика). | (2) |
| знает (мария, английский).  | (3) |
| знает (мария, история).     | (4) |

и сформулирована цель

? знает (мария, X),

причем необходимо найти все решения, удовлетворяющие цели.

Сначала переменная цели X не имеет значения. В этом случае говорят, что переменная неозначена. Неозначенные переменные называют свободными.

Пролог пытается сопоставить цель с фактом (1). Так как первые аргументы несопоставимы, то попытка unsuccessful. На следующем шаге Пролог сопоставляет цель с фактом (2). Значением переменной X становится «информатика», т. е. переменная X означает объект «информатика». Другими словами, эта переменная больше несвободна (связана). Пролог печатает имя объекта, которое обозначает переменная, — «информатика».

Теперь необходимо найти другой объект, который могла бы обозначать переменная X. Это значит, что Пролог должен «забыть» о том, что переменная X связана с объектом «информатика», и снова продолжить поиск с неозначенной переменной X. Таким образом, переменная вновь становится свободной, если цель успешно доказана.

Следующий факт, соответствующий цели, есть (3):

знает (мария, английский).

Переменная X означает объект «английский». Пролог печатает новое значение переменной, «забывает» о том, что X обозначает «английский», и снова продолжает поиск со свободной переменной X. В результате сопоставления цели и факта (4) переменная X означает объект «история».

В нашем примере нет больше информации о том, что знает Мария, поэтому Пролог завершает поиск.

### Использование конъюнкции

Снова вернемся к составлению правил. Попробуем добавить в базу знаний «Дочки-матери» правило, определяющее, кто есть бабушка. Понятно, что Нина является бабушкой Тани потому, что Нина — мама Ольги, а Ольга — мама Тани. Используя понятие переменной, можно сказать иначе: «X является бабушкой для Z, если X является мамой для Y и Y является мамой для Z». Таким образом, условная часть правила состоит из двух отношений, соединенных связкой «и». На Прологе это запишется так:

бабушка (X, Z) :- мама (X, Y), мама (Y, Z).

В этой записи роль связки «и» играет запятая «,».

Здесь мы опять встречаемся с элементом, знакомым нам из темы о базах данных. Фактически правая часть приведенного выше правила является логическим выражением. В нем используется операция логического умножения (и). Если одновременно истинными являются отношения мама (X, Y) и мама (Y, Z), то истинным является и отношение бабушка (X, Z).

В математической логике логическое умножение называется конъюнкцией.

В следующем примере предложите ученикам самим догадаться, какое слово (имя отношения) нужно поставить на место многоточия:

.....(Y, Z) :- мама (X, Y), мама (X, Z)?

Очевидно, здесь пропущено слово «сестра».

Конъюнкцию можно использовать и при формулировке цели (вопроса). Например, к базе знаний «Дочки-матери» можно обратиться с вопросом:

? мама (нина, ольга), мама (нина, тania).

Здесь спрашивается, является ли Нина одновременно мамой Ольги и мамой Тани. Поскольку первое утверждение истинно, а второе — ложно, то в целом на вопрос будет получен отрицательный ответ «нет».

Еще пример сложного вопроса, но уже с переменными: «Существует ли такой курс, который знают Иван и Мария?» Этот вопрос содержит две цели (их можно назвать подделями):

- 1) Какие курсы знает Иван?
- 2) Знает ли эти курсы Мария?

В Прологе эти два вопроса следует объединить, используя конъюнкцию:

? знает (иван, X), знает (мария, X).

Пролог обрабатывает поддели слева направо. Если одна из подделей не может быть доказана, то вся цель является неуспешной (ответ на вопрос отрицательный).

### Как Пролог-система отвечает на вопросы (продолжение)

Теперь рассмотрим алгоритм механизма вывода при получении ответа на вопрос к БЗ, содержащей сложные правила с конъюнкцией.

Дополним базу знаний о Марии и Иване новой информацией. Будем считать, что Мария учится в университете и музыкальном училище. Для изучаемых курсов введем предикаты:

университет\_курс (X)  
музыка\_курс (Y),

смысл которых заключается в следующем: курс X изучается в университете, а курс Y — в музыкальном училище.

Новая база знаний содержит утверждения:

- знает (мария, хор).  
{1} знает (мария, сольфеджио).  
{2} знает (мария, информатика).  
{3} знает (мария, алгебра).  
знает (мария, геометрия).  
знает (иван, X) :- знает (мария, X),  
университет\_курс (X).

университет\_курс (информатика).  
университет\_курс (алгебра).  
университет\_курс (геометрия).

музыка\_курс (хор).  
музыка\_курс (сольфеджио).

И задана цель

? знает (иван, X).

Пролог сопоставляет цель со всеми фактами для факта «знает». Результаты всех сопоставлений неуспешны, так как имя объекта «мария» несопоставимо с именем «иван». Пролог переходит к правилу. Цель и голова правила сопоставимы, так как их переменные свободны. Теперь факты правой части правила становятся подделями. Первой подделью является

? знает (мария, X).

Это новая поддель, поэтому Пролог снова начинает просмотр базы знаний с первого факта для факта «знает» и находит

знает (мария, хор).

Этот факт сопоставим с подделью. Значением переменной X становится «хор». Однако существуют другие утверждения, которые могли быть использованы для доказательства первой поддели. Поэтому Пролог устанавливает указатель отката в точку {1}. С этого указателя Пролог сделает попытку найти другое решение, если вся цель окажется неуспешной.

Так как область действия переменной — это все правило, то вторая поддель есть

**? университет\_курс (хор).**

так как переменная X имеет значение «хор». Все сопоставления этой поддели с фактами базы знаний неуспешны. Поэтому первая попытка доказать цель завершилась неудачей.

Пролог выполняет откат к указателю {1}. Переменная X становится свободной из-за неуспешного вычисления цели. В точке, определяемой указателем отката, Пролог находит утверждение

**знает (мария, сольфеджио)**

и устанавливает указатель отката {2} на следующий факт для факта «знает». Переменная X принимает значение «сольфеджио». Поддель

**? университет\_курс (сольфеджио)**

не может быть доказана, следовательно, доказательство цели снова привело к неудаче. Переменная X освобождается, а для доказательства цели будет сделана следующая попытка. Пролог выполняет откат в точку {2}. Теперь первая поддель сопоставляется с фактом

**знает (мария, информатика).**

X получает значение «информатика», а указатель отката устанавливается в точку {3}. Вторая поддель принимает вид:

**? университет\_курс (информатика).**

Успешное сопоставление этой поддели доказывает цель. Следовательно, ответ на поставленный вопрос формулируется так: «Иван знает информатику». Цель успешно доказана, поэтому переменная X становится свободной и может быть вновь означена при поиске других решений.

### **Особенности конкретных реализаций Пролога**

Пролог для УКНЦ и IBM PC допускает использование следующих символов (алфавит):

- прописные буквы латинского алфавита от A до Z;
- строчные буквы латинского алфавита от a до z;
- цифры от 0 до 9;
- специальные символы ? — ;, () " и др. (остальные специальные символы используются в особых случаях, которые здесь не рассматриваются).

#### **Правила образования имен.**

Имя — последовательность букв и/или цифр, начинающаяся с буквы. Имена отношений и объектов выбираются произвольно. Однако имена должны быть подобраны так, чтобы наилучшим образом отражать смысл тех понятий, которые они представляют. Имена фактов и объектов должны начинаться со строчных букв.

### **Синтаксис Пролога для IBM PC (Турбо Пролог)**

Описание базы знаний и цели на Прологе для IBM PC состоит из трех разделов:

- описания фактов;
- описания цели;
- факты (утверждения).

Ключевые слова predicates, goal и clauses отмечают начало соответствующего раздела:

predicates  
 <описание фактов>  
 goal  
 <цель>  
 clauses  
 <утверждения>

В разделе predicates необходимо указать типы объектов для каждой разновидности фактов. Некоторые из этих объектов могут быть числовыми данными, другие же — символьными строками.

Чтобы факт know можно было использовать для описания утверждений базы знаний, необходимо сделать следующее описание:

predicates  
 know (symbol, symbol)

Это описание означает, что оба объекта факта know относятся к типу symbol. Этот тип является одним из стандартных типов Пролога.

### Стандартные типы Пролога

| Тип данных           | Ключевое слово | Диапазон значений   | Примеры использования     |
|----------------------|----------------|---|---------------------------|
| Символы              | char           | любые символы   | 'a' 'b' 'c'               |
| Целые числа          | integer        | от -32768 до 32767  | -63 89                    |
| Действительные числа | real           | от -1E-307 до 1E308   | -56478 1.25E27            |
| Строки               | string         | последовательность символов   | "Home"                    |
| Символические имена  | symbol         | 1. Имя, первый символ — строчная буква<br>2. Последовательность символов, заключенная в кавычки | flower<br>"Desk" "window" |

Опишем на Турбо Прологе базу знаний и сформулируем цель.

predicates  
 know (symbol, symbol)  
 goal  
 know (X, geometry),  
 write ("Mary know", X).  
 clauses  
 know (mary, informatics).  
 {1} know (mary, english).  
 know (mary, algebra).  
 know (mary, geometry).  
 know (john, What) :- know (mary, What).

Цель, заданная в разделе goal, называется внутренней. Раздел goal может отсутствовать. Тогда Пролог предлагает ввести цель с клавиатуры. Цель, введенная в процессе диалога пользователя с Пролог-системой, называется внешней. После доказательства внешней цели Пролог просит ввести следующую внешнюю цель. Чтобы проигнорировать очередное приглашение системы ввести новую цель, нужно нажать клавишу Esc.

Если цель является внутренней, то Пролог останавливает поиск после первого успешного доказательства цели. Внешняя цель позволяет получить все возможные решения.

### Синтаксис Пролога для УКНЦ

Рассмотрим пример описания базы знаний на Прологе для УКНЦ:

```
{1} know (mary, informatics).
    know (mary, english).
    know (mary, algebra).
    know (mary, geometry).
    know (john, What) :- know (mary, What).
```

После ввода этой базы знаний можно задавать вопросы, ответы на которые Пролог печатает непосредственно под текстом вопроса. Перед вопросом необходимо поставить «?-» . Например,

|                             |         |                         |
|-----------------------------|---------|-------------------------|
| ?-know (mary, informatics). | вопрос: | знает Мэри информатику? |
| yes                         | ответ:  | да                      |
| ?-know (john, algebra).     | вопрос: | знает Джон алгебру?     |
| yes                         | ответ:  | да                      |
| ?-know (john, geography).   | вопрос: | знает Джон географию?   |
| no                          | ответ:  | нет                     |
| ?-know (mary, geometry).    | вопрос: | знает Мэри геометрию?   |
| yes                         | ответ:  | да                      |

Теперь сформулируем вопрос, содержащий переменную:

?-know (join, What).

Пролог ответит

What = informatics

и будет ждать дальнейших указаний. Если нажать на клавишу Return, то поиск других возможных решений будет прекращен. Если вместо этого ввести точку с запятой (;), Пролог продолжит поиск в базе знаний, начиная с места, отмеченного указателем отката {1}.

Следующий ответ:

What = english.

Таким образом, если существует несколько ответов на поставленный вопрос, Пролог-система найдет столько из них, сколько пожелает пользователь.

Используя те же факты, зададим еще один вопрос и рассмотрим диалог пользователя и системы.

|                       |               |   |
|-----------------------|---------------|---|
| ?-know (X, geometry). | вопрос:       | кто знает геометрию?  |
| X=mary                | первый ответ: | Мэри знает геометрию<br>пользователь нажимает <ввод>;             |
| X=join                | второй ответ: | Джон знает геометрию<br>пользователь нажимает <ввод>;             |
| no                    | третий ответ: | нет, так как нет больше информации о тех,<br>кто знает геометрию. |

### Система Пролог-Д

Система Пролог-Д реализована в качестве учебной для школьных КУВТ. Основное ее удобство заключается в использовании в алфавите кириллицы (русских букв). Достаточно подробно Пролог-Д описан в статьях журнала «Информатика и образование» [8].

### Контрольные вопросы и задания

1. На какие две составляющие можно разделить знания человека?
2. Чем кроме знания фактов и правил должен владеть творчески мыслящий человек?
3. Чем отличается база знаний от базы данных?
4. В чем состоит назначение системы Пролог?

5. Что такое экспертная система?
6. Что такое факт в Прологе?
7. Что такое правило в Прологе?
8. Объясните смысл терминов «голова правила» и «тело правила».
9. Что понимают под термином «база знаний» в Прологе?
10. Что такое цель?
11. В чем отличие между первым и вторым типами целей?
12. Пусть имеются утверждения:

Все люди смертны.  
Сократ — человек.

Из этих утверждений следует, что

Сократ — смертен.

Запишем эти утверждения на Прологе:

смертен (X) :- человек (X)  
                  человек (сократ)

и сформулируем цель:

? смертен (сократ)

Описать схему доказательства цели. Каков будет ответ системы на поставленный вопрос?

13. Что обозначают переменные и как они записываются на Прологе?
14. Можно ли формулировать вопросы с использованием переменных?
15. Каков будет результат доказательства цели

смертен (X)

для базы знаний вопроса 12?

16. Объяснить назначение конъюнкции в правилах и целях.

Следующие вопросы можно ставить перед учениками, если на уроках изучался механизм вывода Пролога.

17. Какие переменные называются свободными (неозначенными)?
18. В каких случаях происходит означивание переменной?
19. Когда означенная переменная вновь становится свободной?

20. В каких случаях цель, содержащая конъюнкцию подцелей, не может быть доказана?

21. В каком порядке Пролог просматривает утверждения базы знаний при доказательстве цели?

22. Каково назначение указателей отката?

23. Описать схему доказательства цели с использованием указателей откатов.

### Упражнения

1. Описать базу знаний «Слова», содержащую информацию о синонимах и антонимах. Например, синонимом слова «храбрый» является «смелый». Это утверждение можно записать так:

синоним (храбрый, смелый).

Слова «храбрый» и «трусливый» — антонимы, следовательно, имеет место факт антоним (храбрый, трусливый).

Сформулировать вопросы, назначение которых — нахождение синонима и/или антонима заданного слова.

2. Описать базу знаний «Столицы государств». Например,

столица (Россия, Москва).  
столица (Украина, Киев).

Сформулировать вопросы, содержащие

- имена конкретных объектов;
- одну переменную;
- две переменные.

3. Рассмотреть дерево родственных отношений:



Сформулировать на Прологе следующие вопросы:

- Является ли Федор родителем Ольги?
- Кто является родителем Татьяны?
- Кто дети Ивана?
- Кто является родителем родителя Ольги?
- Кто внуки Ивана?
- Есть ли у Федора и Степана общий родитель?
- Кто чей родитель?

4. Воспользоваться деревом родственных отношений примера 2. Какими будут ответы Пролог-системы на следующие вопросы:

родитель (мария, X)  
родитель (X, мария)  
родитель (иван, X), родитель (X, Y)  
родитель (X, Y)

5. База знаний «Библиотека» содержит информацию:

- о читателях, а именно фамилию и отдел, которым пользуется читатель;
- об услугах, оказываемых различными отделами библиотеки (услуги представим кодами A1, A2, A3 и т. д.).

читатель (иванов, музыкальная\_литература).  
читатель (петров, музыкальная\_литература).  
читатель (сидоров, техническая\_литература).  
читатель (лазарев, периодическая\_литература).  
услуга (музыкальная\_литература, A1).  
услуга (музыкальная\_литература, A2).  
услуга (техническая\_литература, A3).



Дополнить базу знаний новыми фактами и сформулировать вопросы:

- Какие услуги библиотеки может получить читатель Иванов?
- Доступна ли читателю Сидорову услуга А1?

Ответы на какие вопросы можно еще извлечь из базы знаний?

6. Пусть заданы отношения

| Отношение       | Соответствующее утверждение на естественном языке |
|-----------------|---|
| отец (X, Y)     | X является отцом Y                                |
| мать (X, Y)     | X является матерью Y                              |
| мужчина (X)     | X — мужчина                                       |
| женщина (X)     | X — женщина                                       |
| родитель (X, Y) | X является родителем Y                            |
| различны (X, Y) | X и Y различны                                    |

Написать правила, в которых следующие отношения будут головой:

| Отношение               | Соответствующее утверждение на естественном языке |
|-------------------------|---|
| является_сестрой (X, Y) | X является сестрой Y                              |
| дедушка (X, Y)          | X является дедушкой Y                             |
| общие родители (X, Y)   | X и Y имеют общих родителей                       |
| тетя (X, Y)             | X является тетей Y                                |

Описать базу знаний, содержащую описанные выше отношения. Сформулировать вопросы, ответы на которые в явном виде в базе знаний отсутствуют. Объяснить механизм вывода ответов.

7. База знаний содержит информацию об изучаемых курсах в рамках некоторой специализации в учебном заведении, а именно:

- список основных курсов;
- список специальных дисциплин;
- список прочитанных основных курсов;
- для каждой специальной дисциплины — основной курс, который необходимо предварительно прослушать

(специальная дисциплина может быть включена в расписание, если прочитан соответствующий основной курс).

Описать на Прологе базу знаний «Специализация». Сформулировать вопросы:

- Какие основные курсы читаются по специализации?
- Какие специальные дисциплины изучаются в рамках данной специализации?
- Какой основной курс необходимо прослушать для изучения конкретной специальной дисциплины?
- Какие специальные дисциплины могут быть включены в расписание (в расписание можно включить только те дисциплины, для которых прочитаны соответствующие основные курсы)?

8. Описать базу знаний для некоторой предметной области. Сформулировать вопросы, ответы на которые в вашей базе знаний в явном виде отсутствуют.

## Планирование уроков

### Урок 1.

#### Теоретическая часть.

Знания человека как совокупность фактов и правил. Творческое мышление — единство знаний и логики. От реляционных баз данных к базам знаний.

Пролог-система как инструментальное средство для разработки компьютерных баз знаний.

Что такое экспертная система.

**Практическая часть.**

Знакомство с работой на компьютере демонстрационной экспертной системы.

**Урок 2.****Теоретическая часть.**

Основные понятия Пролог-системы: факты, правила, переменные.

**Практическая часть (без ЭВМ).**

Выполнение заданий из упражнений на запись фактов и правил.

**Урок 3.****Теоретическая часть.**

Цели. Синтаксис версии Пролога, используемой в компьютерном классе.

**Практическая часть.**

Знакомство со средой Пролог-системы. Ввод базы знаний из рассмотренных примеров. Постановка целей к БЗ.

**Урок 4.****Теоретическая часть.**

Использование конъюнкции при описании правил и целей. Механизм вывода.

**Практическая часть.**

Описать базу знаний с использованием фактов и правил, содержащих конъюнкцию. Сформулировать цели, состоящие из конъюнкции подделей. Объяснить механизм вывода.

Выдать ученикам задания для индивидуальных работ по теме (использовать задания из приведенных упражнений, а также придуманные учителем). Требуется спроектировать базу знаний и сформулировать к ней достаточное количество вопросов.

**Урок 5.****Практическая часть.**

Выполнение индивидуальных заданий в Пролог-системе на компьютере.

**Урок 6.**

**Заключительное занятие.** Итоговая беседа по всей теме. Завершение выполнения и сдача индивидуальных заданий.

Сдачу заданий можно организовать в форме конкурса на самую «умную» базу знаний.

**Литература**

1. Братко И. Программирование на языке Пролог для искусственного интеллекта. М.: Мир. 1990.
2. Ин Ц., Соломон Д. Использование Турбо Пролога. М.: Мир. 1993.
3. Пугач В. И., Добудько Т. В. Элементы логики и программирования в системе Турбо Пролог. Самара.: СамГПИ. 1993.
4. Информатика: Энциклопедический словарь для начинающих. М.: Педагогика-Пресс. 1994.
5. Каймин В. А., Щеголев Е. А. и др. Основы информатики и вычислительной техники. М.: Просвещение. 1989.
6. Каймин В. А. и др. Статьи в журнале «Информатика и образование». 1989. № 2, 3, 4; 1990. № 2, 6; 1991, № 2, 6.
7. Щеголев Е. А. Школьная информатика и язык Пролог//Информатика и образование. 1993. № 2.
8. Ерохина Е. А. От логики к программированию (Пролог в курсе информатики)//Информатика и образование. 1993. № 2, 5, 6; 1994, № 3, 6.

**Поправка**

В предыдущей статье из данной серии, опубликованной в ИНФО № 5, вместо текста «почта в локальной сети УКНЦ; программа реализована в Пермском государственном педагогическом университете» следует читать: «почта в локальной сети УКНЦ; программа реализована Е. А. Ереминым». Авторы приносят Е. А. Еремину свои извинения.

**М. Г. Эпиктетов,**

кандидат физико-математических наук, МГУ,  
ИнфоМир

**НОВОМИР**

В данной статье дается обзор разделов «Базы данных и электронные таблицы» (VIII класс), «Кодирование информации» (IX класс) и «Информационные модели» (X класс) последней программно-методической разработки НовоМир. В ее состав входит гипертекстовая поддержка, реализованная на платформах DOS и Macintosh. Авторами являются сотрудники механико-математического факультета МГУ и объединения ИнфоМир: А. Г. Кушниренко, А. Г. Леонов, М. Г. Эпиктетов, В. В. Борисенко, Н. А. Подольская, М. А. Кузьменко, С. Б. Ханжин и Б. А. Назаров, работающие под руководством А. Г. Кушниренко.

**Базы данных и электронные таблицы**

Содержание данного раздела составляют две очень важные области применения компьютеров — базы данных и электронные таблицы. Несмотря на то что первоначально компьютеры разрабатывались для проведения сложных расчетов (compute — вычислять), сегодня наиболее массовое применение компьютеров заключается не в вычислениях, а в обработке больших объемов информации. Ознакомление с основными принципами и механизмами такой обработки, несомненно, очень важно.

При изучении баз данных и электронных таблиц в различных курсах традиционно используется «пользовательский подход»: выбирается некоторая производственная система и последовательно, шаг за шагом, изучаются ее команды и способы работы с ней. Такой подход, конечно, годится для подготовки пользователей конкретной системы, однако он не очень хорош при использовании в школьном курсе. Несмотря на то что все системы управления базами данных (равно как и все электронные таблицы) имеют между собой много общего, наличие большого числа довольно сложных технических деталей сильно затемняет общую картину.

Мы попытались отойти от традиционного подхода и рассмотреть основные принципы компьютерной обработки информации, не вдаваясь в технические детали устройства конкретной системы. В частности, в данном курсе используются не реальные производственные системы (которые к тому же надо покупать), а их «облегченные» аналоги: учебно-производственная система КуМир/СУБД и система МикроКальк, разработанные на механико-математическом факультете МГУ. Следует отметить, что, несмотря на слово «учебно» в названии системы, КуМир/СУБД по производственным возможностям (максимальный объем базы данных, скорость поиска и т. д.) ничем не уступает СУБД Paradox фирмы Borland, с которой, кроме того, поддерживается совместимость по формату таблиц. МикроКальк, в свою очередь, является профессиональным текстовым редактором, хотя возможности обработки электронных таблиц в нем не так развиты, как в производственных электронных таблицах.

**Первая глава. «Базы данных»**

Глава начинается с краткого обзора способов хранения информации на компьютере — от файловых систем к информационным системам и базам данных. Далее рассматриваются принципы работы с базами данных: представление базы данных на экране и бумаге (табличное, страничное и смешанное), ввод и редактирование данных, поиск информации по ключу, по значению и по образцу, составление отчетов и т. д. Все рассмотрение ведется на примере двух модельных баз данных: базе данных музыкальных записей и базе данных «Телефонная книга».

Завершается глава изложением некоторых вопросов проектирования баз данных. На примере базы данных электронного депозитария показывается устройство реальной многотабличной базы данных, дается упражнение на создание электронного классного журнала.

## Вторая глава. «Электронные таблицы»

Система МикроКальк, которая используется в данном курсе, отличается от других электронных таблиц тем, что она встроена в редактор текстов МикроМир. Поэтому глава начинается с изучения основных команд редактирования в МикроМире (при этом основное внимание уделяется созданию и редактированию таблиц, нарисованных символами псевдографики).

Знакомство собственно с электронными таблицами включает: изучение основных принципов электронных таблиц (суммирование строк и граф таблицы, расчеты по формулам), вычисление граф по форме, условные вычисления и т. д. Завершается глава изучением дополнительных команд МикроКалька (переключение режимов работы, построение гистограмм, вычисление формул и калькуляторные вычисления).

## Третья глава. «Сортировка и поиск»

Данная глава — это своего рода попытка «заглянуть за кулисы» и продемонстрировать школьникам (не слишком вдаваясь в технические детали) алгоритмы, которые используются (или по крайней мере могут использоваться) в системах управления базами данных при поиске и сортировке. Эта глава не имеет никакой «пользовательской» окраски, однако очень важна для «демистификации» принципов работы баз данных.

На понятных школьникам примерах демонстрируются простейшие алгоритмы сортировки (простое извлечение и пузырьковая сортировка), рассматриваются сортировка слиянием и «быстрая» сортировка, обсуждается вопрос о сложности алгоритмов. В параграфе, посвященном поиску, описаны алгоритмы последовательного поиска в неупорядоченном множестве и двоичного поиска — в упорядоченном, а также использование индексирования.

## Кодирование информации

Изучение способов кодирования и обработки информации неизбежно должно опираться на понятие алгоритма. Вместе с тем, по нашему мнению, в разделах курса информатики, посвященных алгоритмизации, собственно понятие

информации оказывается вторичным, вспомогательным: выполняемый на ЭВМ «его величество Алгоритм» обрабатывает Информацию и, безусловно, выступает главным героем этого процесса. Без ответов остаются вопросы: откуда берется обрабатываемая информация, почему она представлена в той или иной форме, какие свойства объектов реального мира она отражает и т. д.

Однако именно эти вопросы будут выступать на первый план при работе человека XXI века с информационными структурами, при освоении новых методов и инструментов для получения и обработки информации. Поэтому авторы курса постарались сместить акценты с понятия алгоритма на понятие собственно информации, ее кодирования и передачи, а также построения информационных моделей окружающих нас процессов.

## Первая глава. «Повторим программирование»

В этой главе язык программирования не является самостоятельным объектом изучения, а выступает как нотация, система обозначений, на котором ведется обсуждение проблем и их решений. Одновременно язык программирования — это инструмент, обеспечивающий практическую проверку получаемых теоретических решений.

Выбор школьного алгоритмического языка в качестве базисного языка программирования не случаен. Важным является то, что школьный алгоритмический язык погружен в гипертекстовую систему программирования КуМир-Гипертекст. Это позволило снабдить большинство тем курса специально подготовленными гипертекстами, которые предназначены как для организации активной работы учащихся, т. е. для ответов на контрольные вопросы и выполнения других заданий, так и для подачи нового материала.

Материал рассчитан на 10 уроков. За это время школьники должны:

а) изучить или вспомнить основные понятия из курса информатики: алгоритм, величина, ветвление, повторение, присваивание;

б) научиться выражать эти понятия на школьном алгоритмическом языке;

в) научиться основным приемам работы в системе КуМир-Гипертекст.

На первый взгляд кажется, что выполнение пунктов (б) и (в) потребует

запоминания большого количества справочной информации и на это уйдет много времени и сил. Однако это не так.

Во-первых, система КуМир включает развитую гипертекстовую справочную систему и меню для выполнения основных команд, поэтому запоминать придется только то, что нужно каждую секунду.

Во-вторых, все конструкции алгоритмического языка в КуМире вводятся в программу «в готовом виде» с помощью всего двух клавиш. Запоминать в деталях формат этих конструкций необязательно.

В-третьих, если при наборе программы допущена ошибка, КуМир мгновенно отмечает в тексте программы место, где она сделана, а на полях указывает возможную причину ошибки.

Наконец, в гипертекстах по курсу много демонстраций, просмотр которых позволяет мгновенно уловить основные идеи тех или иных команд.

## Вторая глава. «Кодирование информации без компьютера»

Потоки информации в современном обществе все в большей степени переводятся в компьютерную форму, обрабатываются, циркулируют по компьютерным сетям, преобразуются из одной формы в другую. Любая такая информация при необходимости может быть представлена в некоторой стандартизированной, закодированной форме. Как правило, существует несколько форм представления, кодировки одной и той же информации.

Задача главы — познакомить учеников с несколькими формами кодирования, возникшими в докомпьютерную эпоху. Это должно подготовить их к пониманию различных методов кодировки информации на компьютерах, а также подготовить психологически к особенностям этих задач, которые, как правило, не имеют единственного решения.

Изучение «докомпьютерных» способов кодирования информации начинается с обзора способов записи чисел в древности: славянской, римской и вавилонской. Этот материал подготавливает переход к позиционным системам счисления. Большое внимание уделено двоичной системе счисления, алгоритму перевода в двоичную систему и родственным системам счисления (восьмеричной и шестнадцатеричной), как фундаменту для дальнейшего изучения кодирования информации на компьютере. Завершает

тему кодирования чисел раздел о геометрическом представлении чисел и механических вычислителях.

Следующий важный раздел должен сформировать представление о координатах как о способе однозначного задания объекта некоторого заранее описанного класса. Примеры координат на шахматной доске и в кинозале должны помочь ученику избавиться от «математического привкуса» этого термина координаты.

И наконец, завершают главу два важных примера — нотная азбука и кодирование бухгалтерской информации.

## Третья глава. «Кодирование информации на компьютере»

Значение этой темы трудно переоценить. Ее изучение должно снабдить ученика достаточным набором взятых из практики примеров кодирования и перекодирования информации, научить простейшим приемам прикидочных подсчетов, связанных с информацией.

По существу, речь идет о базовых навыках, необходимых для повседневной жизни в XXI веке. Пояснить это можно следующим примером.

Школьник середины XX века должен был уметь решать следующие задачи:

*Сколько рулонов обоев шириной 0,9 метра и длиной 12 метров нужно купить, чтобы отремонтировать квадратную комнату площадью 14 кв. метров и высотой 2,8 метра?*

А школьнику конца XX века — начала XXI помимо этого надо будет уметь решать и задачи такого типа:

*На дискету объема 2,88 Мб требуется записать речь политического деятеля. Сколько минут высококачественной записи речи уместится на дискете?*

### Понятие информации.

Глава начинается с обсуждения понятия информации и определения количества информации. Делается вывод о существовании двух разных мер для определения количества информации. Внешняя, техническая характеристика количества информации в сообщении легко определяется и измеряется и потому является очень важной на практике. Внутренняя (семантическая) мера количества информации должна отвечать интуитивным представлениям о том, что некоторые весьма длинные сообщения в

каком-то смысле несут мало информации или не несут ее вообще.

Оказывается, что единого определения внутреннего количества информации в сообщении, отвечающего этому интуитивному представлению, не существует. А те частные определения, которые удается сформулировать в разных ситуациях, на практике не очень полезны.

### Двоичное кодирование.

Опираясь на уже изученное кодирование чисел с использованием двоичной системы счисления, подробно рассматриваются способы двоичного кодирования самых разных объектов. При двоичном кодировании нельзя использовать ничего, кроме нулей и единиц; нельзя использовать знак «-» при кодировании чисел; нельзя использовать пробел при кодировании текстов (точнее, этот пробел тоже надо закодировать).

Первый пример двоичного кодирования — кодирование чисел. Этот материал во многом уже знаком ученикам, надо только обратить внимание на то, что ячейки имеют фиксированную длину, поэтому приходится дополнять нуль и положительные числа слева нужным количеством нулей. В качестве примера кодирования отрицательных чисел используется дополнительный код, наиболее широко применяемый на практике в настоящее время.

При изучении кодирования текста рассматривается азбука Морзе, как пример кодирования, не являющегося двоичным (несмотря на то что в ней используется всего два символа — точка и тире).

Материал, посвященный кодированию изображений, звуков, музыки и фильмов, не содержит ничего сложного и может быть легко освоен на интуитивном уровне (при использовании имеющейся гипертекстовой поддержки).

### Упаковка и шифрование информации.

Постановки задач здесь очень привлекательны. Цель учителя — не потерять интерес учеников, углубившись в детали алгоритмов. Достаточно, если ученики поймут общую идею каждого из них.

Материал начинается с разбора примеров упаковки с потерей информации и упаковки без таких потерь. Скажем, замена списка учеников с именами и отчествами на список только с инициалами есть пример упаковки с потерей

информации. Ученики могут предложить и собственные примеры.

Рассмотрение шифрования информации начинается с разбора элементарного метода — простой подстановки. Показывается, что этот метод не способен обеспечить требуемой секретности, поскольку очень легко поддается расшифровке. В качестве примера «настоящего» шифрования описываются основные идеи метода «с открытым ключом». В этом методе неожиданна сама идея публикации, казалось бы, секретной информации. Общая схема работы с ключами при шифровке и расшифровке сообщений может быть разобрана без анализа самих алгоритмов, которые, конечно, слишком сложны для учеников.

### Память нашего компьютера.

Заключительный раздел, посвященный структуре организации памяти компьютера, носит обзорный характер. Учитель может выбрать материал по своему вкусу, сосредоточившись либо на технической стороне вопроса (организация шины, структура хранения информации в памяти), либо на экономической (зачем нужны разные виды памяти и что такое кэш-память), либо на общеобразовательной (какие требуют внешние запоминающие устройства и сколько информации они могут «запомнить»).

## Информационные модели

Типичная задача по алгоритмизации в курсе информатики обычно уже описывает, в какой форме и как задана информация, которую должен переработать алгоритм. Задача учащегося — придумать сам алгоритм и составить программу. Это похоже на решение заранее заданного уравнения в курсе математики.

Но в математике помимо изолированных от каких-либо приложений уравнений решаются и текстовые задачи (про поезда, бассейны, станки и т. д.), для чего школьник вначале должен построить «математическую» модель некоторой жизненной ситуации и лишь затем применить известные ему методы для получения ответа с помощью модели. Такие задачи очень важны для включения математики в единую систему образования.

Подобные задачи, в которых вначале нужно придумать «информационную» модель жизненной ситуации, а уж только потом составить программу получения

ответа по этой модели, на наш взгляд, важнее, интереснее и доступнее задачи на составление алгоритмов.

### **Первая глава. «Передача информации»**

Материал этой главы тесно связан с разделом «Кодирование информации». После краткого (на один урок) обсуждения особенностей передачи информации между людьми и повторения некоторых важных аспектов двоичного кодирования информации рассматриваются основные принципы передачи информации между компьютерами.

Одно из основных понятий в компьютерной передаче информации — это понятие протокола, главной особенностью которого является его иерархическое описание. Разумеется, от учеников не требуется разбираться в устройстве какой-нибудь сети в деталях, важно получить общее представление и понять общие принципы.

Далее довольно подробно разбираются принципы работы модема, который является на сегодняшний день наиболее наглядной (и наиболее доступной) частью компьютерных сетей, и методы его программирования. Вместе с тем в книге отмечается, что такой способ передачи цифровой информации не очень удачен и тенденция развития средств телекоммуникации состоит в переходе к цифровой передаче данных.

Следует отметить, что при отсутствии в школе выхода в компьютерную сеть материал придется преподавать «всухую» и он может оказаться для учащихся очень трудным. Однако последующие главы не зависят от изложенного здесь материала, поэтому изучение первой главы можно значительно сократить или перенести на конец года.

### **Вторая глава. «Информационные модели»**

Основной навык, который должны получить школьники после изучения главы, — это умение в простейших ситуациях предложить несколько вариантов информационной модели в зависимости от круга решаемых задач.

Такие задачи привлекательны тем, что очень просты технически и допускают, как правило, несколько равноприемлемых решений. Для каждой задачи по обработке информации, содержащейся в модели, решение может обсуждаться на

двух уровнях. На первом уровне проводятся неформальные обсуждения того, разрешима ли задача вообще, достаточно ли в модели информации для ее решения. Если выясняется, что информации недостаточно, то возникает необходимость в модификации информационной модели, а не в составлении алгоритма. Если же оказывается, что информации достаточно, то можно говорить о переходе к более строгому второму уровню — к составлению формального алгоритма на школьном алгоритмическом языке.

Следует отметить, что задачи, приведенные в учебнике, могут быть заменены другими, предложенными учителем или школьниками. Исключением является материал параграфа «Кодирование геометрической информации», который необходим для поддержки главы 4 учебника и программного обеспечения к этой главе.

Рассмотрение информационных моделей начинается с обсуждения модели кинозала и модели процесса продажи билетов в кинозал. Сначала строится простейшая модель, которая затем расширяется, причем рассматриваются различные расширения модели и обсуждаются их достоинства и недостатки.

Второй пример информационной модели — модель транспортной сети, в которой самые естественные вопросы (например, каково минимальное количество посадок при полете из одного города в другой) могут потребовать довольно сложных алгоритмов. Необходимо, чтобы после рассмотрения простых моделей, где алгоритмы были не очень сложны, школьники столкнулись с ситуацией, когда простые по форме вопросы требуют для ответа сложных алгоритмов.

Далее рассматриваются информационная модель для кодирования геометрической информации, информационные модели поля Робота и рисунка на поле Чертежника. Завершается работа с информационными моделями решением большого количества задач на их построение. Эти уроки можно провести в форме конференции, где школьники делают заранее подготовленные доклады по задачам.

### **Третья глава. «Как устроены компьютерные программы»**

В этой теме разбираются основные принципы создания современных компьютерных программ. Изложение построено на примере трех достаточно

сложных проектов: базы данных «Записная книжка», простого графического редактора и компьютерной игры «Мудрый крот». Основное внимание уделяется не тонкостям программирования, а обсуждению общей структуры программ и использованию в них информационных моделей.

#### Четвертая глава. «Компьютеры в обучении»

Использование компьютеров в обучении излагается на примере системы геометрических построений ПланиМир. Эта система, построенная на базе КуМира, позволяет решать на компьютере задачи на построение из школьного курса геометрии. Достоинствами ПланиМира по сравнению с традиционными построениями циркулем и линейкой являются, во-первых, автоматизация рутинной работы (при минимальных усилиях можно получить аккуратные чертежи) и, во-вторых, возможность трансформации уже

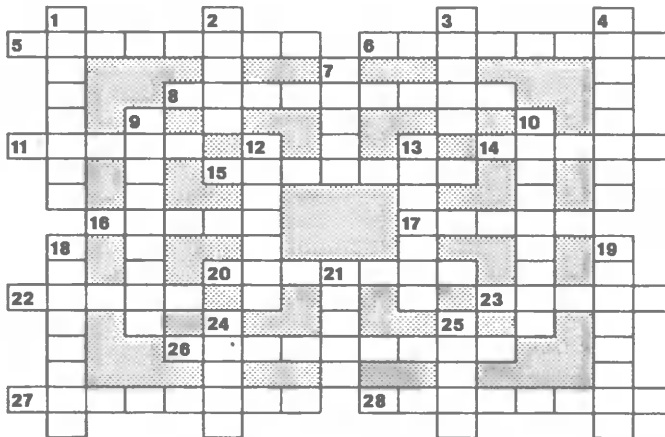
готовых чертежей (с сохранением всех построений).

Таким образом, вначале система ПланиМир рассматривается с точки зрения ее использования — от простейших построений до решения достаточно сложных задач (в частности, с использованием метода геометрических мест). После того как непосредственная работа в ПланиМире будет освоена учениками, можно показать внутреннее устройство ПланиМира. Здесь рассматривается способ представления чертежа в виде алгоритма его построения (этот материал уже обсуждался во второй теме, при рассмотрении информационных моделей геометрической информации). Такой «алгоритмический» подход к геометрическим построениям кроме удобства работы и возможности трансформировать уже построенный чертеж позволяет взглянуть с другой стороны на важное понятие вспомогательного алгоритма, как аналога вспомогательных построений в геометрии.

\* \* \*

## КРОССВОРД

Составитель — В. В. Журавлев (г. Ярославль)



**По горизонтали:** 5. Пионер отечественной кибернетики. 6. Прием исследования, заключающийся в сравнении однотипных результатов, полученных разными путями. 8. Электронное устройство для получения случайных чисел. 11. Одна из основных операций в базах данных. 14. Характеристика кодовой комбинации. 15. Конечное множество символов. 16. Перемещение в одном направлении на одинаковом количестве позиций содержимого в последовательности смежных позиций. 17. Название семейства отечественных ЭВМ второго поколения. 20. Грубое исправление, которое программист обычно считает временным наложением на текст программы первички управления с возвратом. 22. Процесс нахождения в массиве элементов с заданными признаками и их выведение из этого массива. 23. Магнитный элемент, допускающий считывание информации без перемагничивания. 26. Положение системы в заданный момент времени. 27. Рассуждение, формально доказывающее как истинность, так и ложность какого-либо утверждения, высказывания. 28. Коэффициент выражения, устанавливающий количество повторов выполнения последовательности действий, определяемых выражением.

**По вертикали:** 1. Вид проверки. 2. Система прикладных программ, предназначенная для решения задач определенного класса. 3. Положение, истинность которого проверена и доказана практикой. 4. Устный или письменный сообщение, содержащее информацию, которую принимающий информацию должен учитывать в своих действиях. 7. Наименьшая самостоятельная единица речи. 9. Символьный машинно-ориентированный язык программирования. 10. Вытекающий вывод. 12. Выдача чужого произведения за собственное. 13. Максимальное из расстояний в связанном графе. 18. Промежуток времени между двумя сигналами. 19. Американский физиолог, один из основателей нейрокибернетики и один из авторов формальной модели нейрона. 21. Язык спецификаций с графовым представлением потоков управления и данных. 24. Автоматический программно управляемый манипулятор. 25. Американский ученый, крупнейший специалист в области кибернетики.



**Н. П. Радченко,**

*учитель информатики средней школы № 61, Москва*

## ГУМАНИЗАЦИЯ КУРСА ОИВТ

Данная статья содержит методические рекомендации для работы в V—VII классах по программе, разработанной автором и опубликованной в ИНФО № 6 за 1994 г. Напомним, что речь идет о формировании нового, соответствующего национальному образованию смыслового содержания занятий в компьютерных классах. Перед педагогом ставится задача: давая школьнику знания о новых информационных технологиях, обеспечить условия для формирования его эмоционально доброй и богатой личности, эстетически чуткой и нравственно созвучной русскому национальному менталитету. Используемый при изучении информатики интегративный курс «Музыкальное, поэтическое и декоративно-прикладное творчество русского народа в историческом контексте с дохристианских времен до наших дней» позволяет вести обучение и воспитание подростков 10—14 лет в наиболее благоприятной для положительного восприятия форме, так как учитывает возрастные психологические особенности, способствует вовлечению детей в творческий процесс и позволяет индивидуализировать обучение.

Курс ориентирован на раздел ОИВТ «Прикладное программное обеспечение», поэтому приводимые методические рекомендации сгруппированы на имеющие общее применение и специфические для освоения школьниками текстового, музыкального и графического редакторов. Методические рекомендации опираются на наблюдаемое у ребят априорное отсутствие психологического барьера для работы на компьютере. На занятия они идут с радостью, с верой в неограниченные возможности персональных компьютеров, поэтому первостепенная задача преподавателя — сохранить в ребенке эти чувства, не обмануть его ожидания.

Познавательная потребность детей активна, им важно не просто узнать факт — важен сам познавательный процесс, во время которого надо обеспечить условия для духовного саморазвития личности, ориентированной на общечеловеческие ценности.

На занятиях должна быть создана эмоциональная атмосфера положительного восприятия программного материала курса, которое достигается общением с высокохудожественными произведениями (будь то музыка или домашняя утварь) в соответствии с возрастными особенностями ребенка, и задача педагога — научить «ликующему созерцанию», эмоционально подвигнуть к красоте. Таким образом, путь эстетизации учебно-воспитательного процесса — один из методов его совершенствования. При обращении к новым изучаемым темам или объектам, помня о важности первых впечатлений, их целесообразно усилить с помощью эстетической и этической акцентировки.

На уроке необходимо использовать репродукции, слайды, записи музыкальных произведений, изделия народных промыслов.

Эмоциональная целостность урока является условием единства восприятия искусства и созидания, поэтому должна быть организована разнообразная драматургия каждого занятия. (В заключение мы приводим возможное решение для одной из тем.)

Учитель должен обладать чувством и пониманием красоты природы и уметь раскрыть эту красоту (ценную саму по себе и как источник вдохновения народных мастеров), не забывая об экологическом воспитании, так как природа объективно существует такая, каков человек, характер его труда и идеалов.

Формирование устойчивых навыков при работе с текстовым, музыкальным и графическим редакторами должно восприниматься учеником как необходимое средство для творческого самовыражения и, следовательно, не рассматриваться педагогом как самоцель. Мы рекомендуем отказаться от балльной системы оценок за выполненную учеником работу, особенно в начале обучения, так как оценивать в баллах можно только степень усвоения навыков работы с программным обеспечением, а не результат творческого труда. В этом случае уместнее сам факт одобрения и признания.

При обучении по данной программе

ребенок должен стать не просто пользователем ЭВМ, а творцом. Учитель может подсказать возможное техническое решение при реализации на компьютере образца народного творчества, но для развития творческого потенциала предпочтительнее ситуация, когда ученик в диалоге с учителем делает самостоятельный выбор.

Взаимодействие ребят в процессе работы нужно поощрять и организовывать, учить тактичности, терпимости при обсуждении чужих работ и самокритичности при обсуждении своих.

Учитывая неизбежную разницу уровней общего и гуманитарного образования школьников, надо использовать возможности компьютера для индивидуализации обучения. Так, ученикам, дополнительно обучающимся в музыкальной или художественной школе, задания надо подбирать с большей степенью сложности, чтобы для них они были развивающими, а не тормозящими. Подчеркнем также, что дети с более высоким уровнем специального образования должны стать помощниками и консультантами для остальных ребят, лидерами в добровольно образующихся творческих группах.

В системе досуговых учреждений реально наличие разновозрастного состава групп, что нетрудно учесть преподавателю, использующему возможности персонального компьютера для индивидуализации обучения.

Для расширения на школу создаваемой в компьютерном классе социокультурной среды, для открытости курса ОИВТ можно предложить организацию регулярных внутришкольных выставок детских компьютерных рисунков, участие ребят в концертах с использованием компьютерного аккомпанемента, выпуск стенгазет, подготовленных на компьютере и тематически посвященных народному творчеству, издание детьми небольших книг с иллюстрациями. Кроме того, этим достигается положительное авторское отношение к общественному приложению результатов своей работы за компьютером (реализуется принцип «создавать не только для себя, но и для других»), и при этом осваивается коллективная деятельность с использованием НИТ.

Благоприятно влияют на восприятие учащимися программного материала, позволяя им при выполнении на компьютере творческих заданий опереться на личностный опыт поиск детьми этно-

графически характерных произведений народного творчества, способность видеть красоту своего края, занятия вышивкой или лепкой, посещение музеев, походы на природу с учителем информатики.

Программа интегративного курса ориентирована на работу в течение двух учебных лет по одному часу в неделю, например, в V—VI или VI—VII классах (всего 68 часов), или же в течение двух с половиной лет (всего 85 часов), начиная в V классе и заканчивая в первом полугодии VII класса. Школьный курс «Основы информатики и вычислительной техники» распределен на 4—5 лет средней школы, и мы, основываясь на своем опыте, считаем неоправданной спешку при работе по нашей программе. При этом надо учитывать ряд обстоятельств: различные технические возможности компьютеров; замедленное приобретение технологических навыков пятиклассниками на начальной стадии прихода в среднюю школу, что сдерживает возможности творчества; получение специальных знаний о законах построения чертежей, необходимых для успешной работы с темой «Компьютерная графика и живопись», в VII классе; возрастные возможности восприятия вопросов о прошлом и настоящем в российской истории христианства.

На основе принципов народного искусства в художественном обучении ребенка, в развитии его творческого потенциала целесообразно использовать три этапа: повтор, вариация, импровизация. Повтор и вариация — необходимые условия эстетического восприятия и развития детского творчества. Образцом при этом являются произведения традиционных школ народного искусства. Для вариаций и импровизаций в заданиях возможно использование типовых компонентов, заготовок в виде файлов, представляющих мелодию, рисунок, фотографию и т. п., фрагмент которых отсутствует, но который после анализа сохранившейся части и сопоставления отдельных присутствующих элементов мелодии или рисунка ученик может восстановить.

Начало работы с любым новым редактором должно предусматривать часть учебного времени на привыкание ребят к интерфейсу программы, а, овладев минимумом технологических приемов, перейти к самостоятельному творчеству. Поэтому на это время целесообразно на-

метить освоение меню, демонстрацию подготовленных учителем образцов, в том числе в виде файлов с мелодиями и сканированными репродукциями музейных экспонатов. Порядок, в котором из урока в урок происходит обращение к видам народного творчества, обусловлен нарастанием сложности их воспроизведения на ПЭВМ. Учителю необходимо провести адаптацию программы курса, учитывая такую техническую и программные возможности имеющихся компьютеров, руководствуясь прежним критерием: работа должна радовать ребенка, а не ставить перед ним технические неразрешимые задачи. Если программное обеспечение компьютера содержит редакторы, отличающиеся по возможностям, то уместно начать обучение ребят на простейшем, а по мере освоения его и усложнения заданий перейти к редактору с большими возможностями. Например, научившись использовать «Лексикон», можно перейти к Word, после Paintbrush — к Corel Draw.

Начать занятия с использованием компьютеров детям легче всего с текстового редактора, поскольку он связан с привычным представлением информации. Отметим также, что это достаточно эффективный способ освоения детьми клавиатуры. При этом ввод текста с клавиатуры, его отображение на экране дисплея, сохранение на диске, распечатка на принтере — простейший путь для постижения связей между блоками ЭВМ.

Для освоения возможностей текстового редактора мы предлагаем использовать народные поговорки. Это предложение удобной для начинающих длины. Каждую поговорку ребятам надо напечатать и исправить при необходимости все ошибки.

Учиться формировать таблицы при работе с текстом, заполнять их, логически правильно распределять в них текст детям помогут пословицы. Графы таблицы можно назвать: «Что человек наблюдал?», «Что при этом сказал?», «Как мы его поняли?». При заполнении такой таблицы детской фантазии, образному мышлению предоставлен полный простор.

Если в программном обеспечении присутствуют интегрированные среды, то, получая представление об издательской деятельности, подросток должен научиться обмену данными между текстовым и графическим редакторами.

Начинать занятия в компьютерном

классе надо с музыкального редактора, хотя освоение его труднее, чем текстового. Освоение операций, сопутствующих считыванию с МД файлов с мелодиями, можно оформить как интересный детям процесс отгадывания музыкальных загадок. Подготовить для компьютерного аккомпанемента файлы с мелодиями песен ученики смогут уже через несколько занятий в компьютерном классе. Мы предлагаем воспользоваться такой последовательностью: прослушать по очереди работу каждого ученика по команде «дирижера» «компьютерный оркестр», затем, используя такой аккомпанемент, дети начинают петь «про себя», потом так, чтобы слышал сосед, наконец во весь голос. Не бойтесь обнаружить перед учениками отсутствие оперного голоса, отойдите им душой, для них это важнее.

Работа с графическими редакторами наиболее сложна и разнообразна технологическими приемами, поэтому логична для завершающего обучения по программе курса, рассчитанного на полтора-два года. Разнообразие в занятиях вносится широким охватом типов народного творчества, поддающихся воспроизведению на экране дисплея.

Обучая компьютерной графике и живописи, необходимо раскрыть приемы художественного изображения (такие, например, как стилизация, передача объема, расположение на плоскости и в пространстве), развивать пространственное видение.

Эстетическое воспитание средствами природы — проблема общепедагогическая. Подростка не поздно научить видеть красоту симметрии и гармонии в природе, дать представление о единстве материальной и духовной культуры на примере образцов декоративно-прикладного творчества, памятников древнего зодчества.

Программа предусматривает непростую работу по компьютерной живописи в стиле Палеха. Надо помочь ребятам в подборе состава творческих групп для совместной работы над одним рисунком, учитывая разные способности детей к рисованию.

Смена графических редакторов, о которой упоминалось выше, окажется уместной на теме «Работа с палитрой. Кистевая роспись» или «Искусство лаковой миниатюры».

В заключение приведем примеры пяти занятий, посвященных северной вышивке.

Первое занятие — это демонстрация предметов и фотографий музейных экспонатов, размышления об авторах, образе их жизни, мировосприятии, отраженном в их вышивках, о далеких исторических событиях, значении традиций в их творчестве и, наконец, выбор каждым учащимся узора, который ему хотелось бы воспроизвести, используя графический редактор, печать рисунка на принтере. Далее проводим анализ выбранных работ: что изображено на вышивке? Какой предмет она могла украшать? Что можно украсить ею в наше время? Затем задумываемся о композиционном решении автора, использованных декоративных элементах, приемах стилизации. Анализируем возможности нашего помощника — компьютера и сообщаем вывод, каким образом нам удастся решить поставленную задачу. Начинается творчество. Ребята срисовывают со слайда выбранный узор. (Возможен вариант, когда преподавателем для каждого ученика будет заготовлена копия выбранного им узора.) Мы рекомендуем предварительную прорисовку ребенком будущего компьютерного рисунка в первую очередь потому, что чрезвычайно важным является взаимосвязанная работа кисти ребенка и зрительного восприятия. Если рисунок на экране не закончен, то записываем его на диск, что позволит закончить его на следующем занятии. Отметим еще один очень важный момент, благодаря которому ребятам незнакомо сдерживающее их творческое самовыражение чувство беспокойства нарисовать какую-то деталь неудачно, испортить рисунок: современные графические редакторы предоставляют различные средства редактирования рисунка в процессе его создания. Здесь применима известная аксиома К. Д. Ушинского: «Как бы высоко ни был развит отдельный человек, он всегда будет стоять ниже народа». Музейные образцы произведений народного искусства являются для ребят идеалом, с которым надо соотносить свою работу, результат собственного творчества.

На втором занятии завершаем работу, радуемся успеху каждого, потому что идеал невозможно превзойти, а любому ребенку дорог результат его творчества.

Вместе решаем поискать дома предмет, украшенный вышивкой, принести показать в классе.

На третьей встрече, посвященной северной вышивке, мы рассматриваем принесенные из дома образцы, пытаемся установить их возраст, степень сохранности. Ребятам предлагается загадка: «Как мог выглядеть фрагмент вышивки, утраченный во время многовекового хранения?» Они с радостью берутся за воплощение своего варианта отгадки (с диска считывается подготовленная преподавателем с помощью сканера копия древней цветной вышивки, в которой он стер какую-то часть, представив детям как утерянную из-за ветхости). Работа выполняется тоже в цвете и рассчитана на одно занятие. Таким образом, ученики знакомятся с процессом реставрации и постигают ее нравственные основы.

Два заключительных занятия, посвященных импровизации на экране дисплея по мотивам северной вышивки, необычны еще и тем, что к высшей степени раскрытия творческого потенциала ребят ведет звучащая в классе музыка. Мелодия исполняется компьютером и возникает так тихо, что увлеченные работой ребята первое время не замечают ее. Но, прислушавшись, кто-то обязательно начинает «мурлыкать» (ведь мы уже пели под аккомпанемент компьютера русские народные песни, когда работали с музыкальным редактором), постепенно тихонечко поют все, и мы замечаем, что песня и узор рассказывают об одном — о любимой земле. Вспоминаем, что на Руси во время работы всегда пели. Приходим к выводу, что у нас одинаковое мировосприятие с теми, кто в давние времена жил на нашей земле, ведь наша душа во время работы поет, как пела у них, и красивым нам кажется то же, что и им, — та же вышивка, та же земля, та же природа. И вдруг спохватываемся: а та ли природа?

Таким образом, используемый на занятиях компьютер станет средством воспитания и развития творческих способностей ребенка, формирования его личности, дополняющим общепринятые средства и обогащающим педагогический процесс новыми возможностями.

**К. А. Сорокин,**

*учитель математики и информатики средней школы № 297, Москва*

## МУЗЫКАЛЬНЫЕ ВОЗМОЖНОСТИ ЛОГО

В последнее время в методической литературе появляется много статей, посвященных программированию на языке Лого, возможности которого в обучении школьников настолько огромны, что привлекают все большее число преподавателей.

Мой опыт учителя информатики включает в себя работу с учениками на Бейсик-Агат в средах Turbo Basic и Turbo Pascal. Большинство учащихся, работающих в признанных программных средах, испытывают трудности во многом внешнего порядка (работа в среде, компиляция, редактирование). Переход на работу со средой LogoWriter для IBM PC помогает избежать трудностей, научить творчеству в силу своей простоты и наглядности. Уже на первом уроке ученики могут сами составлять простейшие программы в командном режиме, что делает Лого одним из интересных и понятных языков программирования.

В данной статье хочется особо остановиться на музыкальных возможностях Лого.

На первый взгляд музыкальные возможности Лого кажутся бедными: всего одна команда (tone) с двумя параметрами, один из которых задает частоту звучания, а другой — время звучания. И всё же благодаря возможностям черепаховой графики, рекурсивным возможностям можно создавать очень красивые и полезные процедуры, использующие музыкальные эффекты. Рассмотрим три программы, иллюстрирующие возможности создания звуковых эффектов в Лого.

### 1. Ноты

#### Описание программы

При выполнении процедуры `noty!` на экране изображается нотный стан, состоящий из пяти горизонтальных линий, а затем появляются ноты с иллюстрацией их звучания. Для работы программы необходимо создать форму ноты под номером 31, например так, как показано в руководстве по LogoWriter.

```

to noty!
  ct cg
  нотный_стан
  ноты
  if readchar = "q []
end

to нотный_стан
  rg
  ht
  rt 90
  pu
  setpos [-300 50]
  repeat 5 [pd fd 600 pu
            bk 600 rt 90]
            fd 30 lt 90]
end

to ноты
  pu st
  setsh 31
  setx -150 sety -80 pd stamp pu
  tone 262 20
  setx -120 sety -65 pd stamp pu
  tone 294 20
  setx -90 sety -55 pd stamp pu
  tone 330 20
  setx -60 sety -35 pd stamp pu

```

```
tone 349 20
setx -30 sety -25 pd stamp pu
tone 392 20
setx 0 sety -5 pd stamp pu
tone 440 20
setx 30 sety 5 pd stamp pu
tone 494 20
setx 60 sety 25 pd stamp pu
tone 523 20
end
```

Эта программа может быть предложена ученикам в качестве проекта для их самостоятельной работы.

## 2. Пианино

### Описание программы

Простейший музыкальный редактор можно создать с использованием графики и рекурсии.

При выполнении процедуры piano на экране изображаются клавиши фортепиано (процедуры белая\_клавиша, черная\_клавиша) и формируются подписи над черными клавишами (процедура piano) и белыми клавишами (процедура подпись) для соответствия клавиш на фортепиано клавишам на клавиатуре ЭВМ.

Основная процедура music работает рекурсивно. При считывании из буфера клавиатуры символа, звучит соответствующая ему нота, затем опять следует вызов процедуры music и т. д. Выход осуществляется по нажатию символа «x»; все незадействованные клавиши блокируются (последняя команда в процедуре music).

```
to piano
ct rg setc 10 fill ht pu
setx -200 sety -100
repeat 10 [белая_клавиша]
sety 0
setx -210 черная_клавиша
setx -130 черная_клавиша
setx -90 черная_клавиша
setx -10 черная_клавиша
setx 30 черная_клавиша
setx 70 черная_клавиша
setx 150 черная_клавиша
setx 190 черная_клавиша
sety 120
setx -200 label [q]
setx -120 label [e]
setx -80 label [r]
setx 0 label [y]
setx 40 label [u]
setx 80 label [i]
setx 160 label [o]
setx 200 label [p]
подпись "asdfghjkl;
music
ct cg
end

to белая_клавиша
setc 0
pd fd 200 rt 90 fd 40 rt 90 fd 200 rt 90 fd 40 bk 40 pu
setc 1 rt 45 fd 5 pd fill bk 5 rt 45 setc 0 pu
end

to черная_клавиша
setc 0 pd fd 100 rt 90 fd 20 rt 90 fd 100 rt 90 fd 20
bk 20 pu rt 45 fd 5 pd fill pu fd 10 pd fill pu bk 15
rt 45 setc 1
```

end

```
to подпись :f
sety -130
make "l 0
repeat count :f
  [make "i first :f
  make "f bf :f
  setx -190 + :l
  setc 13 pd
  label :i
  make "l :l + 40 pu]
sety -145 setx 5
pd label [выход - x]
pu
end
```

```
to music
make "y readchar
if :y = "a [tone 247 3 music]
if :y = "s [tone 262 3 music]
if :y = "d [tone 294 3 music]
if :y = "f [tone 330 3 music]
if :y = "g [tone 349 3 music]
if :y = "h [tone 392 3 music]
if :y = "j [tone 440 3 music]
if :y = "k [tone 494 3 music]
if :y = "l [tone 523 3 music]
if :y = ";" [tone 587 3 music]
if :y = "q [tone 233 3 music]
if :y = "e [tone 277 3 music]
if :y = "r [tone 311 3 music]
if :y = "y [tone 370 3 music]
if :y = "u [tone 415 3 music]
if :y = "i [tone 466 3 music]
if :y = "o [tone 554 3 music]
if :y = "p [tone 622 3 music]
ifelse :y = "x [stop] [music]
end
```

Программа может быть использована при работе с младшими школьниками для иллюстрации возможностей Logo.

### 3. Команда play

#### Описание программы

Для проигрывания мелодий в языке Бейсик имеется оператор play. Предлагаемая программа позволяет включить такой же оператор в язык LogoWriter:

play "строка

Строка составляется по следующим правилам:

Op , n=0, 1, 2, 3 — выбор октавы: 0 — большая, 1 — малая, 2 — первая, 3 — вторая.

Ln , n=8, 4, 2, 1 — выбор длительности ноты: 8 — восьмая, 4 — четверть, 2 — половина, 1 — целая.

Pn , n=8, 4, 2, 1 — пауза: 8 — восьмая, 4 — четверть, 2 — половина, 1 — целая.

Ноты: C — до; D — ре; E — ми; F — фа; G — соль; A — ля; H — си; # диэз; — (знак минус) — бемоль.

Примечание:

1) в начале строки должен указываться номер октавы;

2) две одинаковые ноты одной длительности лучше записывать с явным указанием их длительностей: l8g18g.

to play :st

```

play! :st 0 0
end

to play! :st :v :s
ht
make "n first :st
make "st bf :st
if not :st = " [make "n1 first :st]
if or :n1 = "# :n1 = "-" [make "n se :n :n1]
if :n = "o [make "k first :st]
if :n = "l [make "v first :st]
if :v = 8 [make "v 2]
if :v = 4 [make "v 4]
if :v = 2 [make "v 8]
if :v = 1 [make "v 16]
if :n = "p [make "s first :st]
if :s = 8 [make "s 1]
if :s = 4 [make "s 2]
if :s = 2 [make "s 4]
if :s = 1 [make "s 8]
if :k = 0 [nota0 :n :v :s]
if :k = 1 [nota1 :n :v :s]
if :k = 2 [nota2 :n :v :s]
if :k = 3 [nota3 :n :v :s]
make "z count :st
ifelse :z = 0 [stop] [play! :st :v :s]
end

```

```

to nota0 :y :v :s
if :y = "C [tone 65 :v]
if :y = "D [tone 73 :v]
if :y = "E [tone 82 :v]
if :y = "F [tone 87 :v]
if :y = "G [tone 98 :v]
if :y = "A [tone 110 :v]
if :y = "H [tone 123 :v]
if or :y = [C #] :y = [D -] [tone 69 :v]
if or :y = [D #] :y = [E -] [tone 78 :v]
if or :y = [F #] :y = [G -] [tone 92 :v]
if or :y = [G #] :y = [A -] [tone 104 :v]
if or :y = [A #] :y = [H -] [tone 117 :v]
if :n = "p [wait :s]
end

```

```

to nota1 :y :v :s
if :y = "C [tone 131 :v]
if :y = "D [tone 147 :v]
if :y = "E [tone 165 :v]
if :y = "F [tone 175 :v]
if :y = "G [tone 196 :v]
if :y = "A [tone 220 :v]
if :y = "H [tone 247 :v]
if or :y = [C #] :y = [D -] [tone 139 :v]
if or :y = [D #] :y = [E -] [tone 156 :v]
if or :y = [F #] :y = [G -] [tone 185 :v]
if or :y = [G #] :y = [A -] [tone 208 :v]
if or :y = [A #] :y = [H -] [tone 233 :v]
if :n = "p [wait :s]
end

```

```

to nota2 :y :v :s
if :y = "C [tone 262 :v]
if :y = "D [tone 294 :v]
if :y = "E [tone 330 :v]
if :y = "F [tone 349 :v]
if :y = "G [tone 392 :v]
if :y = "A [tone 440 :v]
if :y = "H [tone 494 :v]

```



```

if or :y - [C #] :y - [D -] [tone 277 :v]
if or :y - [D #] :y - [E -] [tone 311 :v]
if or :y - [F #] :y - [G -] [tone 370 :v]
if or :y - [G #] :y - [A -] [tone 415 :v]
if or :y - [A #] :y - [H -] [tone 466 :v]
if :n - "p [wait :s]
end

```

```

to nota3 :y :l :s
if :y - "C [tone 523 :l]
if :y - "D [tone 587 :l]
if :y - "E [tone 659 :l]
if :y - "F [tone 698 :l]
if :y - "G [tone 784 :l]
if :y - "A [tone 880 :l]
if :y - "H [tone 988 :l]
if or :y - [C #] :y - [D -] [tone 554 :l]
if or :y - [D #] :y - [E -] [tone 622 :l]
if or :y - [F #] :y - [G -] [tone 740 :l]
if or :y - [G #] :y - [A -] [tone 830 :l]
if or :y - [A #] :y - [H -] [tone 932 :l]
if :n - "p [wait :s]
end

```

```

to kreser
play "o2l8e-dcl2gl1gl8gfe-l2e-ddl8de-fl4gl1gl8fga-l1hplay "p8o3l8co2h-o3co2l2h-a-
o3l8co2h-a-l4a-l1g
play "o3l8co2ge-l2dl4a-l8gl2fp8l8e-fl1g
play "o3p8l8co2h-o3co2l2h-a-o3l8co2h-a-l4a-l1g
play "o3l8co2ge-l2dl4a-l8gl2fp8l8de-l1c
end

```

```

to mishka
play "o1l4ho2c#l2dl4del2f#l4gf#o3l2dc#o2ho3l4dc#o2l2h
play "o2l4agl2f#l4agle
play "o2p8l4hl4hl2hl4gel2c#l4egl2f#a#ho3l4c#o2hl2c#l4ag
play "o1l2a#o2l4gf#o1l4h
end

```

### Структура программы

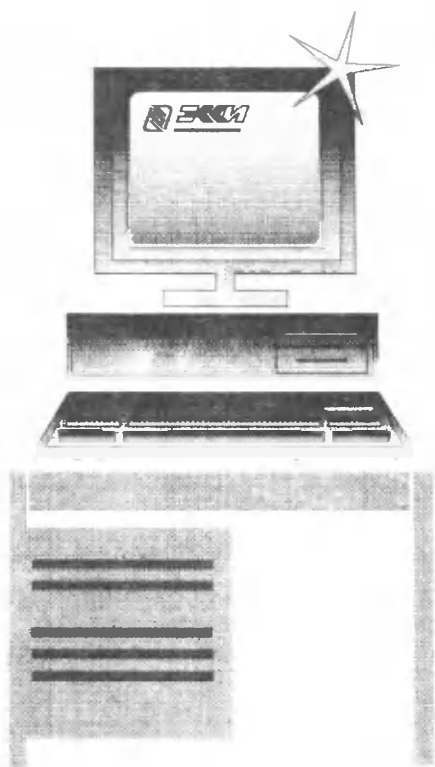
Процедура play вызывает процедуру play! с тремя параметрами: в play! передается строка и два нулевых параметра (это начальные значения длительности и паузы).

В процедуре play! осуществляется разбиение строки и анализ каждого символа. Эта работа проводится рекурсивно до окончания строки. В зависимости от номера октавы управление передается соответствующим этим октавам процедурам: nota0, nota1, nota2, nota3. В эти процедуры передается значение ноты, длительности и паузы.

В конце программы для примера приведены две мелодии: детская песня «Крейсер "Аврора"» и песня об олимпийском Мишке. Команду play можно применять для оформления графических программ музыкальными эффектами.

*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.

**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 IBM, УКНЦ, БК

107005, Москва, Волховский пер., 11  
Тел./факс: (095) 265-62-65  
Тел.: (095) 267-70-58

 **ЭКСИ**  
АОЗТ Фирма "ЭКСИ"

# ПЕДАГОГИЧЕСКИЙ ОПЫТ

**О. Я. Гольдшлаг,**

*председатель методического объединения учителей информатики школы-лицея № 1605,  
Москва*

## ГОТОВИМСЯ К ВЫПУСКНЫМ ЭКЗАМЕНАМ

В школе-лицее № 1605 в физико-математических классах информатика является профилирующим предметом начиная с VIII класса. Преподавание ведется по двум программам: четырехгодичной (306 часов — по 2 часа в неделю в VIII—IX классах и 3 часа в неделю в XI классе) и интенсивной (170 часов — по 2 часа в неделю в X классе и 3 часа — в XI). Сложившаяся практика обусловлена дополнительным набором учащихся в VIII и X физико-математические классы. Базовый объем знаний согласуется с программой непрерывного курса обучения информатике. Компенсировать разницу в часах, отведенных на изучение курса, удается за счет применения различных методических приемов, факультативных занятий, перераспределения акцентов на самостоятельное изучение некоторых тем курса.

Ежегодная проверка знаний проводится в форме устного экзамена. Задания, предлагаемые школьникам, позволяют оценить степень усвоения ими фундаментальных основ информатики и приобретенных практических навыков работы на ПЭВМ типа IBM PC. Билет содержит теоретический вопрос и задачу. Для выполнения задания школьнику отводится 40 минут. Объем программы должен не превышать 20 строк, но в то же время она должна включать защиту от некорректных исходных данных, комментарии, вывод результатов.

Предлагаем вашему вниманию экзаменационные вопросы по информатике, предлагавшиеся в 1994/95 учебном году для учащихся одиннадцатых физико-математических классов.

### Билет № 1

1. Аппаратура и архитектура ПЭВМ типа IBM PC. Назначение и основные

характеристики компонентов и комплекса в целом.

С помощью средств Norton Commander (NC) определить параметры оперативной и дисковой памяти.

### 2. Задача на тему «Циклы».

В видеоигре игрок получает 50 очков за сбитый самолет, 100 — за сбитую ракету и 200 — за сбитый спутник. Выведите на экран число очков игрока, который сбил в первой игре  $a$  самолетов,  $b$  ракет и  $c$  спутников, а во второй —  $d$  самолетов,  $e$  ракет и  $v$  спутников.

### Билет № 2

1. Информация, логическая организация информации на диске.

Используя средства NC, создать файл `d:\user\exzamen\var1.txt` и записать в него назначение NC, название и назначение функциональных клавиш (F1—F10) основного меню NC.

2. Задача на тему «Циклы, условные операторы».

Дано  $n$  натуральных чисел  $1..n$ . Найти сумму четных чисел  $s1$  и нечетных чисел  $s2$ .

### Билет № 3

1. Двоичная система счисления. Единицы измерения информации — бит, байт, килобайт, мегабайт — и их соотношение.

Расчитать, сколько дискет емкостью 360 Кбайт необходимо для размещения учебного пособия, содержащего 420 страниц (70 символов в строке, 80 строк на странице).

### 2. Задача на тему «Массивы, циклы».

Для заданных чисел  $a$  и  $b$  в таблице  $t[1..n]$  из  $n$  действительных чисел подсчитать число элементов, попадающих в

интервал  $[a..b]$ . Использовать переменные:  $i$  — для счетчика индексов;  $k$  — для хранения результата.

#### Билет № 4

1. Объяснить причину использования двоичной системы счисления в вычислительной технике.

Записать четное и нечетное пятиразрядные двоичные числа. Перевести их в десятичную систему счисления.

Логические операции и их таблицы истинности.

2. Задача на тему «Циклы».

Расчитать таблицу значений функции  $f(x) = \sqrt{\sin x}$  при изменении  $x$  от 0 до  $\pi$  с шагом 0,1  $\pi$ .

Разработать алгоритм, составить блок-схему, написать программу.

#### Билет № 5

1. Логическая организация информации на диске. Понятия пути, файла, каталога, подкаталога. Правила образования имен файлов и каталогов, маски файлов.

Средствами NC создать файл  $d:\user\exzamen\var2.txt$ . Записать в него все команды DOS, позволяющие работать с каталогами. Скопировать информацию в файл  $d:\user\var3.txt$ , исходный файл удалить.

2. Задача на тему «Циклы, условные операторы».

Подсчитать, сколько чисел натурального ряда будет кратно произвольно задаваемой величине при условии задания с клавиатуры: значения последнего числа ряда; общего делителя.

#### Билет № 6

1. Этапы решения задач на ЭВМ.

Средствами NC создать файл  $d:\user\exzamen\var3.txt$ . Записать в него все варианты использования команды DOS `copy`. Скопировать информацию в файл  $d:\user\exzam\inf\var3.txt$ . Удалить средствами DOS все созданные файлы и каталоги второго и третьего уровней.

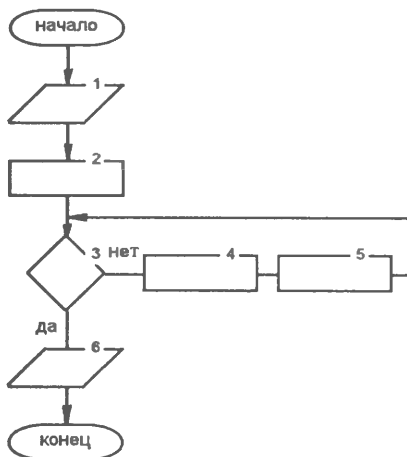
2. Задача на тему «Циклы, условные операторы».

Дано  $n$  натуральных чисел  $1..n$ . Найти сумму четных чисел  $s1$  и нечетных чисел  $s2$ .

#### Билет № 7

1. Понятие алгоритма. Его свойства, способы записи.

Придумать текст задачи, которая реализуется следующим алгоритмом:



2. Задача на тему «Внутренние команды MS DOS».

Средствами DOS создать каталог  $d:\user\exz$ , в который поместить файл  $vnkom.txt$ , содержащий название и назначение внутренних команд MS DOS для работы с каталогами. Вывести содержимое этого файла на экран.

#### Билет № 8

1. Понятия программы, алгоритмического языка.

Структура программы на Паскале.

Написать программу, не содержащую ни одной переменной и выводящую на экран основные возможности MS DOS.

2. Задача на тему «Циклы».

Стоимость проезда в одну сторону до ближайшей станции 2000 рублей. Билет до следующей станции стоит в два раза больше, до третьей в три и т. д. Билет может быть полный — 100% стоимости, льготный — 50% и детский — 25% полной стоимости.

Разработать алгоритм построения таблицы стоимости билетов всех видов для проезда до каждой из станций, начиная со следующей и до десятой включительно, нарисовать блок-схему алгоритма и написать программу, выводящую таблицу на экран монитора.

**Билет № 9**

1. Текстовые редакторы. Назначение и основные возможности. С помощью текстового редактора Лексикон напечатать текст, разделенный на два абзаца. В первом абзаце перечислить функции DOS 5.0, а во втором записать основные внутренние и внешние команды DOS 5.0.

2. *Задача на тему «Функции».*

Определить наибольший модуль вектора для пары векторов, координаты которых заданы в трехмерном пространстве.

Составить алгоритм вычисления модуля трехмерного вектора, нарисовать блок-схему алгоритма и написать соответствующую ему функцию. Написать программу, выполняющую решение задачи с помощью составленной функции.

**Билет № 10**

1. Электронные таблицы. Назначение и основные возможности.

Создать таблицу в файле `d:\user\exzam\vyrexz.cal` и на ее основе проиллюстрировать виды диаграмм, реализованные в SuperCalc 4.

2. *Задача на тему «Процедуры».*

Вычислить произведение заданного пространственного вектора на заданное число. Составить алгоритм умножения вектора на число, нарисовать его блок-схему и написать соответствующую ему процедуру. Написать программу, выполняющую решение задачи с помощью вызова составленной процедуры.

**Билет № 11**

1. Электронные таблицы. Назначение и основные возможности. Понятия формулы, текста.

Основные математические функции, реализованные в SuperCalc 4. При ответе использовать таблицу, содержащую сведения об учениках XI параллели, хранящуюся в файле `d:\user\exzam\uch11.cal`.

2. *Задача на тему «Условные операторы».*

В коробке с красками лежат тюбики с номерами: 1 — красный, 2 — зеленый, 3 — желтый, 4 — черный, 5 — белый, 6 — синий, 7 — коричневый. Написать программу определения номера тюбика по задаваемому с клавиатуры цвету или выдавать сообщение «Такой краски нет».

**Билет № 12**

1. Основные этапы загрузки DOS.

Назначение командных (batch) файлов DOS 5.0. Объяснить правила их написания и назначение команд DOS 5.0 на примере файла `autoexec.bat`.

2. *Задача на тему «Циклы».*

Получив краткосрочную ссуду в банке, в течение первой недели необходимо вернуть занятые деньги плюс 10% этой суммы. В течение каждой следующей недели возвращаемая сумма увеличивается на 10% по сравнению с предыдущей неделей (так называемые «сложные проценты»). Разработать алгоритм, нарисовать блок-схему и написать программу расчета возвращаемой суммы в срок от 1 до 12 недель.

**Билет № 13**

1. Структура программы на Паскале. Описание переменных и констант, последовательность размещения их в блоке описания данных, правила написания имен. Проиллюстрировать на примере.

2. *Задача на тему «Массивы, условные операторы».*

Подсчитать сумму элементов таблицы  $\{1..n\}$  из  $n$  действительных чисел, которые могут быть как положительными, так и отрицательными. В зависимости от полученного результата вывести одно из сообщений: «сумма больше нуля» или «сумма меньше нуля». Нарисовать блок-схему и написать программу. Использовать переменные:  $i$  — для счетчика индексов;  $s$  — для хранения результата.

**Билет № 14**

1. Порядок решения задачи в среде Турбо Паскаль. Возможности встроенного в нее редактора. Проиллюстрировать примерами.

2. *Задача на тему «Функции».*

Подсчитать число пробелов в строке текста. Разработать алгоритм, составить блок-схему, написать соответствующую функцию.

Написать программу, использующую составленную функцию для определения количества пробелов в строке, введенной с клавиатуры.

**Билет № 15**

1. Операторы Паскаля, присваивающие значения переменным, зависи-

мым и независимым от исходных данных программы. Проиллюстрируйте на примерах.

Типы данных в Паскале и их характеристики.

2. *Задача на тему «Массивы. Типы данных».*

В экзаменационной ведомости фиксируются фамилии учеников, номера билетов, оценки в виде цифр и прописью. Составить подобную ведомость для группы из 10 учащихся и на ее основе сформировать список учащихся, сдавших экзамен на «отлично». Строка списка имеет следующий вид:

Иванов билет № 13 5 (отлично)

### Билет № 16

1. Типы данных в алгоритмическом языке Турбо Паскаль. Перечислить характеристики стандартных типов данных (объем оперативной памяти, диапазон изменения, точность вычисления, разрешенные операции). Арифметические операции Паскаля для данных типа integer, real. Приоритеты арифметических операций. Преобразование числовых данных, необходимые для этого в роенные математические функции.

2. *Задача на тему «Синтаксис операторов Паскаля».*

Дана последовательность операторов, вычисляющих факториал числа n.

```
k:=1; f:=0;
while k<n do
f=f*k
k:=k+1
trunc(k)
```

Найти содержащиеся в ней ошибки и недостающие операторы.

### Билет № 17

1. Данные логического типа. Логические операции. Правила записи сложных условий. Условные операторы Паскаля; блок-схемы их работы.

2. *Задача на тему «Электронные таблицы. Математические функции».*

С помощью электронной таблицы составить накладную для продажи 5 сортов мороженого, в которой указать стоимость и количество порций каждого сорта. На основании накладной подсчитать выручку за 30 дней продажи. Определить самое дешевое мороженое и мороженого какого сорта было продано больше всего.

### Билет № 18

1. Оператор выбора case, блок-схема его работы. Сходство и различие с условными операторами. Привести пример с ключом выбора типа char.

2. *Задача на тему «Линейные алгоритмы».*

Четыре человека пообедали в ресторане. Официант подал им счет по x рублей каждому. Они решают оставить официанту чаевые в размере 15% счета. Определите сумму чаевых.

### Билет № 19

1. Операторы цикла в Паскале. Тело цикла, параметры цикла. Блок-схемы работы операторов цикла.

Написать программу, иллюстрирующую работу цикла while...do.

2. *Задача на тему «Внутренние команды MS DOS».*

Средствами MS DOS создать каталог d:\user\exz, в который поместить файл vnkcmd.txt, содержащий название и значение внутренних команд MS DOS для работы с файлами. Объединить содержимое этого файла с файлом c:\config.sys.

### Билет № 20

1. Логические операторы цикла. Блок-схемы их работы. Сравнительная характеристика их работы.

Написать программу сложения четных чисел от 2 до 20 с помощью этих конструкций.

2. *Задача на тему «Текстовый редактор. Фрагменты текста, поиск, замена».*

В текстовом редакторе написать четверостишие, которое сохранить в файле d:\user\exzam\stih.txt. Добавить к нему четверостишие из файла d:\user\exzam\var\stih1.txt. Заменить все буквы «а» на «!!», а затем третью «!!» заменить на «а». Название четверостишия написать курсивом. Информацию сохранить в файле d:\user\exzam\var1\rez.txt.

### Билет № 21

1. Оператор цикла по пересчету, блок-схема его работы.

Привести примеры цикла с использованием переменной цикла, от которой тело цикла не зависит.

2. *Задача на тему «Электронные таблицы, построение графиков».*

С помощью электронной таблицы построить график функции  $f(x)=\log(x)$  на интервале от 1 до 100. График сопроводить поясняющими надписями. Результат сохранить в файле `d:\user\exzam\var1\rez.cal`.

**Билет № 22**

1. Правила организации вложенных циклов.

Написать программу, вычисляющую значения  $y=3x+z$  при всех значениях  $x=\{0,5; 1; 1,5; 2,0; 2,5\}$  и при всех значениях  $z=\{1; 2; 3; 4\}$ .

2. Задача на тему «Текстовый редактор. Работа с окнами, фрагментами текста».

В текстовом редакторе написать рассказ о выпускных экзаменах, красную строку начать с позиции 12, левую границу абзаца установить в позицию 6, а правую — на позиции 60. Вторую фразу повторить в конце рассказа еще раз. Название подчеркнуть. Рассказ сохранить в файле `d:\user\exzam\exz.txt`.

**Билет № 23**

1. Функции и процедуры.

Правила их описания и вызова в Паскале.

Локальные и глобальные переменные.

2. Задача на тему «Электронные таблицы. Формулы, текст».

С помощью электронной таблицы составить сводную таблицу, отражающую результаты сдачи экзаменов по информатике в физико-математических классах. Шапка таблицы имеет следующий вид:

| Номер<br>классв | Пяте-<br>рок | Чет<br>верок | Троек | Двоек | Неявкв |
|-----------------|--------------|--------------|-------|-------|--------|
|-----------------|--------------|--------------|-------|-------|--------|

В последней строке подвести общий итог. Результат сохранить в файле `d:\user\exzam\var1\rez2.cal`.

**Билет № 24**

1. Понятие массива, индекса, элемента массива. Правила их описания в Паскале.

Назначение, описание и обращение к данным типа `record`.

2. Задача на тему «Текстовый редактор. Графический режим».

В текстовом редакторе в окне номер 4 на зеленом фоне черными символами в виде таблицы записать информацию о странах Европы: название страны, название столицы, государственный язык. Информацию сохранить в файле `d:\user\exzam\var1\rez1.txt`.

**Билет № 25**

1. Типы файлов, используемые в Паскале для организации данных. Правила описания их в программах.

2. Задача на тему «Электронные таблицы, построение диаграмм».

С помощью электронной таблицы построить диаграмму, отображающую количество учащихся в физико-математических классах VIII—XI параллелей. Диаграммы сопроводить поясняющими надписями. Результат сохранить в файле `d:\user\exzam\var1\rez.cal`.

**Билет № 26**

1. Управляемые параметры дисплея, работающего в графическом режиме. Основные графические примитивы.

2. Задача на тему «Циклы».

Вычислить значение функции  $y=x+x+a$  на отрезке  $[-10,+10]$  с шагом 0,75.



*It's staying anyway!*

Объединение ИнфоМир  
предлагает  
органам управления образованием  
Генеральную лицензию на  
распространение на территории  
республики, края,  
области, города, района  
рекомендованного Министерством  
образования в качестве основного  
учебного материала для курса  
информатики  
**учебно-методического  
комплекта КуМир**

Всегда в свободной продаже

КуМир-Гипертекст для MAC и DOS, МикроМир для MAC, DOS и Windows  
ПаскальМир и НовоМир для DOS, КуМир для УКНЦ, Корвета и Ямахи

**Заявка на приобретение программных средств**

Я хочу приобрести: \_\_\_\_\_

Тип ЭВМ: \_\_\_\_\_ Счет выслать по адресу: \_\_\_\_\_



Оригинальную продукцию Объединения ИнфоМир можно приобрести  
только у официальных дилеров. Остерегайтесь подделок!

Заявки направляйте по адресу: 103051, Москва, Садовая-Сухаревская,  
дом 16, комната 9. Редакция журнала "Информатика и образование"

Для получения полного каталога вложите конверт с маркой

*Пользователи глобальной сети Internet  
могут получить демонстрационные версии  
программ Объединения ИнфоМир  
URL: <ftp://tech.math.msu.su/pub/InfoMir>  
Имя для входа: *anonymus*  
В качестве пароля следует указать Ваш  
электронный адрес*

Copyright Объединение ИнфоМир © МСМХСУ  
ИнфоМир, лого Объединение ИнфоМир, КуМир,  
КуМир-Гипертекст, ПаскальМир, НовоМир,  
МикроМир - торговые марки Объединения ИнфоМир





# НОВЫЕ ПРОЕКТЫ

**В. М. Буданов,**

*кандидат физико-математических наук, научный сотрудник Института механики Московского государственного университета им. М. В. Ломоносова*

**К. В. Савицкий,**

*младший научный сотрудник Института механики Московского государственного университета им. М. В. Ломоносова*

## СОРЕВНОВАНИЯ РОБОТОВ ВО ФРАНЦИИ

С 22 по 27 мая 1995 г. в небольшом французском городе Ферте-Бернар проходил II Международный фестиваль наук и технологий, в котором принимала участие и успешно выступила команда сотрудников и студентов МГУ.

Фестиваль был задуман как зрелищное соревнование мобильных роботов и прошел впервые в 1994 г. в новом культурном центре Ферте-Бернара. Организатором фестиваля является компания «Decouverte et communication», которая пользуется активной поддержкой Министерства высшей школы и науки Франции, а также министра почты и телекоммуникаций Франсуа Фийона. Следует отметить, что проведение фестиваля требует серьезных финансовых затрат и было бы невозможно без значительной спонсорской поддержки таких крупных промышленных фирм, как Филипс, Франс Телеком, Рено. Целями фестиваля являются поощрение творчества в области новых технологий, расширение научных контактов внутри Франции и за ее пределами, привлечение молодежи к перспективным областям техники.

Возникновение фестиваля и его поддержка на разных уровнях, в том числе и на государственном, конечно, не являются случайными. Робототехника активно внедряется в промышленное производство, многие технологические операции сейчас немыслимы без применения высокоточных, надежных и «умных» ро-

бототехнических систем. Без преувеличения можно говорить о том, что робототехника является одним из важнейших направлений научно-технического прогресса, в котором проблемы механики и новых технологий соприкасаются с проблемами искусственного интеллекта. Соответственно, необходимы специалисты, способные обслуживать, производить и разрабатывать такую технику. С этой точки зрения фестиваль выполняет две важные функции — привлекает школьников и существенно повышает качество подготовки специалистов в высших учебных заведениях. В конечном счете факт возникновения фестиваля свидетельствует о внимании, с которым во Франции относятся к проблеме сохранения и роста своего интеллектуального потенциала. На первом фестивале были представлены мобильные роботы трех типов: колесные, шагающие и микророботы. На втором фестивале колесные роботы были разбиты на два класса: свободный и монотип, при этом участникам в классе «монотип» в начале учебного года были предоставлены одинаковые базовые наборы, содержащие платформу размером 60×80 см, колеса и двигатели. На основе этих элементов студенческие команды должны были в течение учебного года создать мобильные роботы, способные выполнять весьма непростые действия, чтобы справиться с зада-

чами, сформулированными в регламентах соревнований.

По регламенту соревнований робот класса «монотип» должен иметь возможность двигаться по полигону вдоль полосы шириной 5 см. Полигон представляет собой площадку из чередующихся квадратов черного и белого цвета со стороной 2 м. На черные квадраты нанесена белая полоса, а на белые квадраты — соответственно черная. Робот должен собирать шарики от французского бильярда красного цвета и не брать шарики другого цвета, начинать движение из-за старта и остановиться после финиша, а в случае разрыва полосы найти ее продолжение по катафотному отражателю, установленному на высоте 1 м над точкой продолжения трассы. Свободный класс от «монотипа» отличался тем, что в нем могли принимать участие произвольные колесные роботы. Также был несколько усложнен регламент соревнований.

Начало нашим контактам с организаторами фестиваля было положено в 1994 г., когда в отдел общей механики Института механики МГУ пришло приглашение на участие в фестивале, которое свидетельствовало об известности за рубежом наших работ в области шагающих машин. Однако приглашение пришло слишком поздно и мы не смогли в полной мере принять участие в первом фестивале. Тем не менее двое сотрудников нашего коллектива, А. В. Ленский и Е. В. Гурфинкель, побывавшие там, показали видеофильм о нашей шагающей машине и продемонстрировали маленькую и простую, но оригинальную шагающую модель, сделанную много лет назад А. В. Ленским. Эта модель вызвала живой интерес публики и участников. В сентябре 1994 г. было получено приглашение принять участие во втором фестивале. Так как большого опыта в области создания колесных машин мы не имели, то было принято решение о том, что команда МГУ примет участие в соревнованиях в классах «монотип» и шагающих роботов. Для разработки колесного робота в сентябре—октябре была сформирована группа из 12 студентов

кафедр теоретической механики и прикладной механики и управления механико-математического факультета МГУ. Определался также «тренерский коллектив» из сотрудников института, в который входили Е. А. Девянин, Е. В. Гурфинкель, А. В. Ленский, В. М. Буданов и К. В. Савицкий. Активное участие в формировании концепции мобильного робота принял заведующий кафедрой теоретической механики академик Д. Е. Охоцимский. В работе помогали и другие сотрудники отделов общей механики и вычислительной техники института. Для эффективной организации работы Институт механики МГУ выделил специальное помещение. Нашлись и спонсоры — фирма «Стрингер» подарила компьютер Notebook. Основные расходы по пребыванию нашей команды во Франции взял на себя оргкомитет соревнований, финансовую поддержку оказал ректорат МГУ.

На еженедельных дискуссиях, проводимых сотрудниками ИМ МГУ вместе со студентами, решались вопросы создания робота, обсуждалось и критиковалось все, невзирая на авторитеты. В ноябре — декабре студенты уже разделились на группы, каждая из которых начала заниматься определенной частью работы. Одни начали писать моделирующие программы, которые впоследствии должны были стать основой для реализации основных алгоритмов управления, другие — разрабатывать необходимую электронику, третьи — создавать макеты основных узлов. Главной идеей при разработке робота было то, что для успешного участия в соревнованиях машина должна быть как можно более «умной». Поэтому система управления была создана на базе компьютера Notebook, который через электронные блоки получает информацию от всех датчиков (слежения за полосой, локаторов поиска отраженного сигнала от катафота, датчиков определения наличия шариков в механизме захвата, сигнал от видеокамеры о цвете шара и т. д.) и передает управляющие сигналы на все исполнительные устройства (двигатели колес, двигатель



**FESTIVAL  
INTERNATIONAL  
DES SCIENCES  
ET TECHNOLOGIES**

**É-BERNARD**

**28 MAI 95**

**MONDE  
MOBILE  
MÉDI**



Фото 1



Фото 2



Фото 3

Фото 5



Фото 4



Франция, Ферте-Бернар, весна 95-го... Придумал и организовал этот фестиваль известный французский тележурналист М. Юган (фото 2, справа). Может быть, отчасти, поэтому журналисты, съехавшиеся из разных стран, чувствовали себя на фестивале комфортно. От парижского аэропорта Шарль-де-Голль до Ферте-Бернара организаторы фестиваля доставили нас на «Рено» со скоростью 180 км/час («Рено» — традиционно один из спонсоров фестиваля).

Представьте небольшой, очень уютный, необыкновенно чистый, весь в цветах и зелени, с озером и пляжем городок, в котором живет около 10 000 жителей, и в огромном супермаркете которого наименований только продовольственных товаров значительно больше. Фестиваль робототехники — событие для Ферте-Бернара. Построены специальные павильоны (фото 1); министр почты и телекоммуникаций Франсуа Фийон (фото 2, слева) посетил фестиваль; оригинальные и умные роботы, которые при-



Фото 6



Фото 7

везла команда из России (фото 3, 4), получившая один из призов (фото 5); ежедневная газета; заседания пресс-клуба, транслирующиеся по национальному телевидению; спектакль на площади перед городской ратушей....

Самое главное, что можно посмотреть (а иногда и потрогать!), как и из чего собираются роботы (фото 6, 7), на которых во время выступления некоторые команды одевали «красивую одежду» (фото 8). На соревнования приходили семьями и болели за любимейшего робота по-настоящему. Чемпионы (фото 9, 10), которые выполнили все сложные задания жюри, по нашему мнению, были вполне достойны быть экспонатами музея Ферте-Бернара. Может быть на следующем фестивале в 1996 году так и будет?

Фото 8

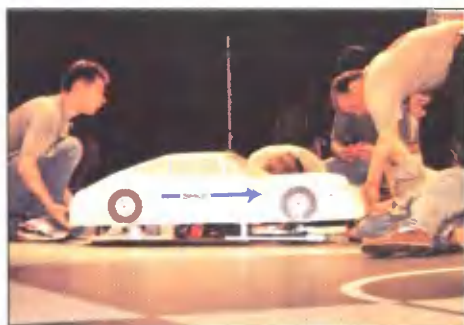


Фото 9

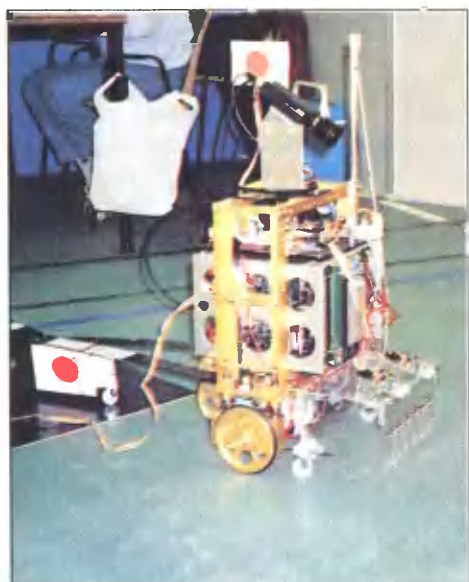


Фото 10





Один из идеологов и основателей направления, которое можно перевести на русский язык как «педагогическая робототехника» — профессор Ля-Манского университета М. Виве. Он был так-



же одним из руководителей команды своего университета на фестивале в Ферте-Бернаре. Красивая желто-синяя «Yellow Submarine», изящно выполняющая программу под мелодию известной песни Битлз, увезла заслуженный приз.



Совсем коротко о сути направления можно сказать так. В лабораторных условиях моделируются элементы, фрагменты, комплексные узлы промышленного производства (например, цех по сборке автомобилей). Использование новых информационных технологий в сочетании с учебными роботами позволяет моделировать, изучать и отрабатывать (в том числе и в реальном времени) различные производственные процессы и ситуации. Такие комплексы можно использовать и для решения целого ряда учебных задач из смежных дисциплин. Написаны учебные программы, методические пособия, готовятся к защите диссертации.

Кроме этого, М. Виве является главным редактором журнала «Sciences et techniques éducatives». Этот журнал был основан во Франции в 1994 году и призван освещать все аспекты «педагогической робототехники» (или, по-другому, образовательной инженерии). Это новое направление педагогической науки начинает развиваться во многих странах. Существуют различные европейские проекты, связанные с этим направлением, — ESPRIT, DELTA, COMETT и другие. В России в этой области работают специалисты Московского института новых технологий (ИНТ).

В Канаде в 1996 году состоится международный симпозиум по «педагогической робототехнике», и М. Виве является одним из его организаторов.

В № 2/1995 журнала «Информатика и образование» была опубликована статья М. Виве «Образовательная инженерия». Мы собираемся и дальше освещать эту тематику.



устройства захвата шариков, двигатели локаторов и даже динамики, способные воспроизводить музыку, записанную в память компьютера). Было написано программное обеспечение, реализующее все необходимые функции для работы машины и позволяющее изменять поведение аппарата и адаптировать его к различным условиям (например, изменению регламента соревнований). В результате рассмотрения различных вариантов отдельных частей робота, многократных проб, исследования возможностей использования тех или иных как промышленно выпускаемых элементов, так и разработанных в лаборатории постепенно создавался робот.

К концу марта определилась четверка лучших студентов, которым в мае предстояло принять непосредственное участие в соревнованиях. Ими стали А. В. Безнос, А. В. Валиев (IV курс), О. И. Подкопаев (III курс), И. Н. Каленова (II курс). Как обычно, самым жарким временем оказался месяц перед отъездом во Францию, месяц в течение которого из отдельных частей надо было собрать «боеспособный» робот и отладить управляющие программы. Кроме этого необходимо было придумать и сделать корпус робота, дизайн которого должен быть не только красивым, но и оригинальным, ведь эти соревнования, кроме всего прочего, еще и красивое зрелище и болеть за участников придет целый город (хоть и маленький).

Почти сразу после того, как машина была собрана, ее предстояло разобрать, упаковать, доставить в Шереметьево, пройти таможенную (что оказалось делом отнюдь не простым). Только уже находясь в самолете, мы смогли почувствовать себя относительно спокойно. После четырех часов полета мы приземлились в Парижском аэропорту Шарль-де-Голль. Здесь мы впервые почувствовали четкость работы организаторов фестиваля. На двух легковых машинах и микроавтобусе прямо из аэропорта нас привезли в Ферте-Бернар. Высокий уровень организации фестиваля проявлялся буквально во всем. Для нормальной работы

команд были организованы электрифицированные рабочие места (боксы) в вестибюле культурного центра и спортзале близлежащей спортивной школы. Командам оказывалась разнообразная техническая помощь, от изготовления механических деталей до предоставления измерительной и вычислительной техники. Специально для проведения соревнований был построен временный шатер (его называли «шапито») с площадью 10 на 20 метров, окруженной трибунами. В культурном центре в течение всей недели проходили лекции и обсуждения, посвященные важнейшим научным и техническим проблемам. Все было буквально напичкано телевизионной аппаратурой, на десятках телевизоров отражались текущие события. Велась трансляция по местному телевидению, участников соревнований узнавали и приветствовали на улицах города. Ежедневно издавался «Журнал фестиваля».

Если в прошлом году в фестивале принимали участие в основном команды из Франции, то в этом году Ферте-Бернар приветствовал участников многих ведущих университетов мира — из Франции, США, Канады, Японии, России, Германии, Южной Кореи, Португалии. Несмотря на то что между большинством участников соревнований существовал языковой барьер, все чувствовали себя вполне уверенно и комфортно.

В течение недели команды собирали и испытывали свои машины на тренировочной и основной трассах, приспосабливали технику с учетом освещенности и контрастности трассы, придумывали новые алгоритмы управления, доводили до рабочего состояния то, что не успели сделать дома. Команды свободно общались, помогали друг другу. Надо сказать, что не было ни одной команды, у которой бы не возникали проблемы с ее роботом. Абсолютно всем приходилось работать буквально днем и ночью с перерывами на еду и весьма непродолжительный сон. Вполне естественным было зайти в три часа ночи и застать четыре-пять команд, корпящими над своим детищем. Днем в

боксах было значительно оживленнее. За работой команд в непосредственной близости могли наблюдать посетители, местные и приехавшие специально, поодиночке и семьями. Зрители вели себя очень корректно, например, если где-то было указано, что заходить запрещено, никому и в голову не приходило это делать. В целом было ощущение, что фестиваль является праздником для всего города.

К концу недели были проведены отборочные и полуфинальные соревнования, в результате которых из 16 роботов класса «монотип» к финалу были допущены 5, в том числе и робот команды МГУ. Во время полуфинальных и финальных соревнований трибуны были полны, зрители активно поддерживали участников, каждый удачный проход трассы вызывал бурные овации. Было удивительно, что из такого, казалось бы, сугубо технического мероприятия организаторы смогли сделать столь увлекательное представление.

В субботу, 25 мая, фестиваль посетил министр почты и телекоммуникаций Франции, который выступил перед участниками и зрителями с продолжительной речью, в которой подчеркнул важность этого мероприятия и обещал свою поддержку в дальнейшем. В тот же день были проведены финальные соревнования мобильных роботов, по результатам которых первое место в классе «монотип» заняла команда из французского города Нант, а в свободном классе — команда университета Цукуба (Япония). В последний день, в воскресенье, состоялись показательные выступления четырех пред-

ставленных на фестиваль шагающих машин и соревнования микророботов. Последние представляют собой миниатюрные колесные машинки с микропроцессорным управлением и датчиками, фиксирующими наличие препятствий. Соревновались они в скорости и «разумности» при движении по лабиринту.

Шагающие машины были столь различны по габаритам, принципам организации движения, уровню подготовленности (например, машина из Парижской лаборатории робототехники впервые «пошла» непосредственно перед выступлением), что какие-либо соревнования между ними были нереальны. Поэтому все ограничилось демонстрацией их возможностей. Наша шагающая машина продемонстрировала способность маневрирования и ходьбы по лестнице. Нужно отметить, что ходьба по лестнице была исходно запланирована в регламенте, но остальные машины этого сделать не смогли. В тот же день состоялось вручение призов. Команда МГУ получила два специальных приза в классе «монотип», главный приз по шагающим машинам — поездку в Монреаль в следующем году, а в целом была удостоена гран-при зрителей.

Успешное выступление нашей команды в новой для нас области мобильных роботов, большой интеллектуальный запас нашей машины, не реализованный в полной мере из-за массы технических проблем, вселяют в нас уверенность в дальнейших успехах в будущих фестивалях, которые являются прекрасной школой для студентов.

**От редакции.** Те, кто заинтересовался этой публикацией, могут обратиться в редакцию за дополнительной информацией.



**Е. А. Захарова,**

*старший научный сотрудник лаборатории экранных искусств Исследовательского центра эстетического воспитания Российской академии образования*

## РАЗВИВАЮЩИЙ ЭКРАН

Прошедший в начале октября российско-британский семинар «Медиаобразование-95» обнаружил существенный для российского образования факт: наличие двух направлений в отечественном медиаобразовании, которые, как «два берега одной реки», текли параллельно, не пересекаясь. Первое — связано с компьютеризацией школы; второе — берет свое начало с 20-х годов нашего столетия — от кинообразования, которое сегодня наконец завоевало свои позиции. Об этом свидетельствует выпуск утвержденной Министерством образования РФ программы «Основы экранной культуры».

Встретились «физики» и «лирики» и поняли, что сотрудничество их неизбежно, так как для современного ребенка давным-давно существует единый экран, как друг, как способ познания и общения с миром, как естественный язык, как нормальная социокультурная среда. Этот единый экран состоит из экранов кино, видео, телевидения и экрана компьютера.

Как обнаружили в своих исследованиях ученые лаборатории экранных искусств Исследовательского центра эстетического воспитания Российской академии образования, восприятие, мышление и способности современного ребенка формируются под мощным воздействием экрана. Мышление ребенка сегодня как бы «телекинематографично», оно более визуализировано, чем мышление детей «доэкранный» эпохи. Оно, если можно так выразиться, «мультипликационно» — дети легко «считывают» с экрана динамичный, насыщенный аудиовизуальный образ, пишут «сценарии», рисуют многочисленные раскадровки-сериалы.

А поскольку наша школа всегда страдала излишней вербализацией обучения, то сегодня, наконец, наступает для медиа настоящая эпоха расцвета — эпоха, позволяющая сделать обучение более наглядно-образным, более очеловеченным и доступным для ребенка. И в этом школе поможет экран кино, телевидения, видео, компьютера.

С этих позиций письмо МО РФ от 04.01.95 № 5/11 «О создании медиацентров образовательных учреждений» весьма дальновидное и своевременное решение, которое, как мы надеемся, позволит развиваться плодотворному сотрудничеству двух ветвей медиаобразования.

На российско-британском семинаре был представлен ряд программ британских и российских авторов, с одной из которых мы хотим познакомить вас сегодня.

Программа «Развивающий экран» Е. А. Захаровой представляет интерес не только как возможность приобщения учащихся к основам экранной культуры, но и с точки зрения особого акцента в данной программе на развитие ребенка средствами и на материале искусства анимации, в том числе и компьютерной, и собственного творчества детей в области анимации. При ознакомлении автора программы с компьютерными технологиями выяснилось, что многие творческие задания, игры и упражнения, предлагаемые данной программой, успешно адаптируются к возможностям компьютера. Разработкой компьютерных вариантов этой развивающей программы автор и занимается сегодня.

Программа «Развивающий экран» рассчитана на обучение и развитие учащихся гуманитарных школ, гимназий,

лицеев, классов эстетического профиля, поэтому в ней расширены и углублены разделы, связанные с историей искусств и культуры, развитием образного восприятия и мышления, воображения, фантазии и творческих способностей учащихся средствами и на материале экранных искусств (кино, телевидения, видео, компьютерной анимации и пр.). Синтетическая природа искусства экрана и особые потенциальные возможности восприятия и воздействия экрана на ребенка позволили создать комплексную, интегрированную программу, в которой сделана попытка решить ряд проблем детского развития, не решаемых другими программами.

В структуре программы 4 блока.

Блок I — «Введение в экранный художественный образ» — ставит своей целью развитие образного восприятия и мышления, специфического именно для общения с экранными искусствами, развития эмоциональной и эстетической отзывчивости на экранный образ.

Блок II — «Экранный язык, речь, повествование» — расширяет задачи блока I и предлагает ребенку осваивать различные виды экранного повествования. На первом этапе это мифологический тип повествования на экране, который интересен не только своей принадлежностью к классической культуре, но и тем, что показывает истоки многих современных экранных жанров, в том числе «фэнтези» и «мыльной оперы».

Блок III — «Из истории культуры. Истоки зранных искусств» — позволяет не только проследить с учащимися зарождение экранных искусств, их «генетические» корни в истории культуры, начиная с первобытного искусства, но и ввести ребят в контекст культуры, пройдя своеобразный курс истории искусств и культуры на увлекательном, доступном для учащихся материале.

Блок IV — «Развивающий экран» — блок интенсивного развития ребенка на материале и средствами экранных искусств, который явился результатом трехлетней работы автора программы в детском доме. Блок предлагает различные виды деятельности для развития эмпатии, различных видов восприятия (визуального, цветового, слухового, музыкального и др.), различных видов памяти, интеллектуального и образного мышления, развития воображения, фантазии, творческих способностей детей.

Материал программы не ориентирован жестко на определенный возраст. Он скорее предлагает определенные уровни, ступени развития. Поэтому первую часть программы можно начать не только в I, II классах, но и позже. Программа оснащена подробными методическими разработками.

Желающие получить программу «Развивающий экран» могут заказать ее по телефону: (095) 201-72-66 (вторник, четверг), *Захарова Елена Ангреевна*.

# ИНФОРМАТИКА В МЛАДШИХ КЛАССАХ

**Е. П. Коляда,**

*учитель информатики гимназии № 87, г. Саратов,  
член проблемной лаборатории кафедры педагогики естественных факультетов  
СГУ им. Н. Г. Чернышевского*

## РАЗВИТИЕ ЛОГИЧЕСКОГО И АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ УЧАЩИХСЯ I КЛАССА

Информатика изучается в средней школе в основном в старших классах. Однако в настоящее время проведено много экспериментов по введению курса информатики в младших и средних классах. Это объясняется, во-первых, важной ролью курса в развитии логического и алгоритмического мышления. Во-вторых, освоив компьютер в младших классах, ученики затем смогут применять его для изучения других предметов.

По пути введения информатики с I класса пошли и в гимназии № 87 г. Саратова. Занятия проводились 1 раз в неделю (34 урока в год). Класс делился на две подгруппы по 12 человек. Одна подгруппа класса занималась с психологом, другая — информатикой. Согласно санитарным нормам, за компьютером ученики проводили 15 минут. Каждый урок строился следующим образом: первая половина отводилась для объяснения

фундаментальных понятий, решения логических задач и алгоритмических этюдов, повторения и закрепления пройденного, вторая — практической работе за компьютерами.

При работе в младших классах очень важно учитывать возрастные особенности детей, знать психологическую карту каждого ребенка, поэтому необходим тесный контакт учителя со школьным психологом.

При построении курса мы учитывали опыт американских, болгарских и российских исследователей, а также разработки учителей-новаторов, с которыми мы познакомились на курсах повышения квалификации учителей информатики, проводимых Саратовским областным институтом усовершенствования учителей.

С самого начала были определены цели курса, а также требования к знаниям и умениям учащихся:

| ЦЕЛИ  | ТРЕБОВАНИЯ К ЗНАНИЯМ И УМЕНИЯМ УЧАЩИХСЯ<br>Учащиеся должны:  |
|---|--|
| 1   | 2  |
| 1. Формирование представления об информации и о способах работы с нею | 1. Иметь представление о многообразии форм информации.<br>2. Знать об основных способах работы с информацией (хранение, передача, обработка).<br>3. Уметь приводить примеры информационных процессов.<br>4. Уметь приводить примеры обработки информации |

| 1   | 2  |
|---|--|
| 2. Знакомство с устройством компьютера и областями его применения                     | <ol style="list-style-type: none"> <li>1. Иметь представление об ЭВМ как универсальной машине для работы с информацией.</li> <li>2. Знать основные составляющие части компьютера и их назначение.</li> <li>3. Знать основные сферы применения компьютеров.</li> <li>4. Уметь приводить примеры использования компьютеров в окружающем мире.</li> <li>5. Иметь представление об истории развития ЭВМ</li> </ol>   |
| 3. Умение работать с готовыми компьютерными программами                               | <ol style="list-style-type: none"> <li>1. Знать о назначении и применении графического редактора.</li> <li>2. Уметь работать в графическом редакторе: создавать простые изображения с помощью имеющихся шаблонов.</li> <li>3. Знать о назначении и применении текстового редактора, основные приемы редактирования (удаление, замена, вставка символа).</li> <li>4. Уметь набрать и отредактировать простой текст.</li> <li>5. Знать о назначении и применении музыкального редактора.</li> <li>6. Уметь работать в музыкальном редакторе: набирать простые мелодии.</li> <li>7. Уметь работать с несложными программами по русскому языку и математике.</li> <li>8. Уметь пользоваться игровыми программами, клавиатурными тренажерами</li> </ol> |
| 4. Развитие логического мышления  | <ol style="list-style-type: none"> <li>1. Иметь представление о логических «не», «и».</li> <li>2. Уметь выделить признак, по которому произведена классификация.</li> <li>3. Уметь выделить общий признак двух групп предметов.</li> <li>4. Уметь решать задачи на поиск недостающих фигур путем применения зрительного и мыслительного анализа.</li> <li>5. Уметь решать задачи, в которых осуществляется одно традуктивное умозаключение (от единичного к единичному или от общего к общему)</li> </ol>  |
| 5. Формирование и развитие навыков алгоритмического мышления                          | <ol style="list-style-type: none"> <li>1. Иметь представление об алгоритме как о последовательности действий, которые приводят к решению задачи.</li> <li>2. Уметь описывать повседневную деятельность в виде алгоритмов.</li> <li>3. Находить и устранять ошибки в линейных алгоритмах.</li> <li>4. Уметь составлять простые алгоритмы переливания, перевозки, переформирования составов.</li> <li>5. Иметь представление о блок-схемах.</li> <li>6. Уметь составлять блок-схему: линейной последовательности действий, с одним ветвлением.</li> <li>7. Иметь представление об исполнителе, его среде обитания, системе команд, отказах</li> </ol>  |
| 6. Развитие активного творчества с использованием современных компьютерных технологий | Приобрести навыки самостоятельного творчества в графическом, музыкальном, текстовом редакторах   |

**ПРОГРАММА КУРСА**  
**«Основы информатики и вычислительной техники»**  
**для I класса (34 часа)**

| № п/п | Тема   | Логические и дидактические игры и задачи              | Освоение готовых программных продуктов               | Количество часов |
|-------|--|---|--|------------------|
| 1     | 2  | 3   | 4  | 5                |
| 1     | Правила поведения и ТБ в компьютерном классе. Первое знакомство с ЭВМ. Из чего состоит ЭВМ | —   | Компьютерная игра «Докер»                            | 1                |
| 2     | Что такое информация. Передача, хранение, обработка информации                             | Игры «Отгадай предмет», «Телефон»                     | Клавиатурный тренажер «Курсор»                       | 1                |
| 3     | Обработка информации.  | Сказочная база данных                                 | Клавиатурный тренажер «Клавиа»                       | 1                |
| 4     | Кодирование информации   | Игра «Шифровка»                                       | Клавиатурный тренажер «Ворон»                        | 1                |
| 5     | Декодирование информации   | Игра «Разведчик»                                      | Диалог с компьютером: клавиатурный тренажер «Привет» | 1                |
| 6     | История развития средств вычислительной техники  | —   | Компьютерная игра «Докер»                            | 1                |
| 7     | Что умеет делать компьютер. Знакомство с функциональными клавишами                         | Игра «Аукцион»  | Программа по математике                              | 1                |
| 8     | Как работает компьютер   | Игра «Вычислительная машина», задача «Поиск девятого» | Программа по математике                              | 1                |
| 9     | Понятие о координатной плоскости   | Игра «Расположи предмет»                              | Контролирующая программа по русскому языку           | 1                |
| 10    | Понятие исполнителя. Управление исполнителем   | —   | Робот (КуМир)  | 1                |
| 11    | Понятие исполнителя. Управление исполнителем   | Задачи с неполным условием                            | Робот (КуМир)  | 1                |
| 12    | Система команд исполнителя, среда обитания, отказы   | Какая фигура лишняя и почему? Какая фигура следующая? | Исполнитель Квадратик (Роботландия)                  | 1                |
| 13    | Понятие алгоритма  | Алгоритм приготовления завтрака, алгоритм перевоза    | Перевозчик (Роботландия)                             | 1                |
| 14    | Понятие алгоритма  | Алгоритм переливания                                  | Переливашка (Роботландия)                            | 2                |

| 1  | 2   | 3   | 4   | 5 |
|----|---|---|---|---|
| 15 | Понятие алгоритма   | Алгоритмы — черные ящики (угадывание алгоритмов)                                | Буквоед (Роботландия)   | 2 |
| 16 | Понятие блок-схем   | Составление блок-схем линейных алгоритмов                                       | Ханойская башня (Роботландия)                                   | 2 |
| 17 | Блок-схемы. Ветвление   | Составление блок-схем алгоритмов с ветвлением                                   | Контролирующая программа по русскому языку                      | 2 |
| 18 | Понятие о машинной графике, ее использовании. Назначение графического редактора | Решение логических задач. Алгоритмы формирования составов                       | Графический редактор Раскрашка (Роботландия)                    | 2 |
| 19 | Музыкальный редактор, его возможности и назначение                              | Алгоритмы формирования составов   | Музыкальный редактор Шарманщик (Роботландия)                    | 2 |
| 20 | Текстовый редактор, его возможности и применение                                | Решение логических задач  | Текстовый редактор Микрон (Роботландия)                         | 1 |
| 21 | Основные правила редактирования   | Алгоритмы формирования составов   | Текстовый редактор Микрон (Роботландия)                         | 1 |
| 22 | Интуитивное понятие множества. Логическое «НЕ»                                  | Решение логических задач  | Текстовый редактор Микрон (Роботландия)                         | 1 |
| 23 | Логическое «И». Выделение общего признака группы                                | Решение логических задач  | Текстовый редактор Микрон (Роботландия)                         | 3 |
| 24 | —   | Логические игры «Какого цвета твоя шапка», «Угадай название цветка на карточке» | Отчетные творческие работы в текстовом и графическом редакторах | 2 |

Данный курс создавался из расчета использования класса УКНЦ со следующей программной поддержкой:

- пакет программ «Роботландия»;
- пакет программ «КуМир»;
- клавиатурные тренажеры «Ворон», «Клавиа»;
- логическая компьютерная игра «До-кер»;
- пакет программ «Русский язык»;
- пакет программ по математике для проверки навыков устного счета;
- настольная детская игра «Путешествие в мир логики».

Если в вашей школе нет некоторых из перечисленных программ или у вас не класс УКНЦ, то для программной поддержки можно взять любой клавиатурный тренажер, любые текстовый, графический и музыкальный редакто-

ры, самим составить простые программы, контролирующие навыки устного счета в пределах 100, и программу по русскому языку.

Остановлюсь подробно на разработке темы «Логические задачи». При разработке этой темы очень важна подборка задач. Они должны быть посильными и интересными, только тогда можно найти у ребят живой отклик. Логические задачи представляют собой упражнения в мыслительной деятельности. Мышление младших школьников в основном конкретное, образное, поэтому при решении таких задач необходимо применять наглядность. В зависимости от особенностей упражнений в качестве наглядности применяются рисунки, чертежи, краткие условия задач, таблицы и др.

Приведем ряд з а д а ч, которые с успехом решают первоклассники:

1. Художник хотел заполнить фигурами каждую из девяти клеток игрового поля. Но передумал и нарисовал фигуры только в восьми клетках, оставив девятую свободной. Какой рисунок хотел нарисовать художник в свободной клетке?

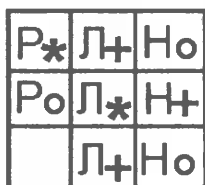


Рис. 1.

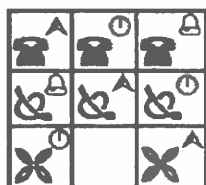


Рис. 2.

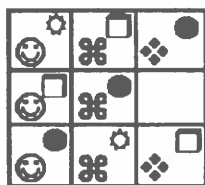


Рис. 3.

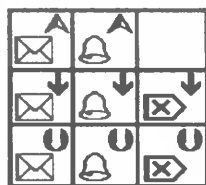


Рис. 4.

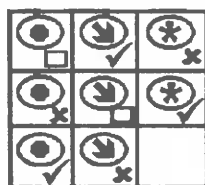


Рис. 5.

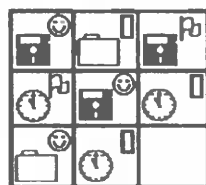


Рис. 6.

2. Какая из данных фигур лишняя? Почему?



Рис. 7.

3. Какой рисунок должен быть в свободной клетке?

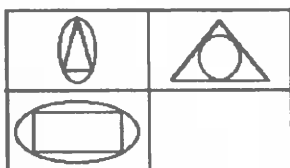


Рис. 8.

4. Выберите нужную фигуру из четырех пронумерованных.

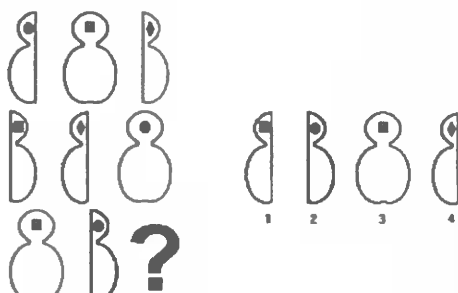


Рис. 9.

5. Выберите нужную фигуру из шести пронумерованных.

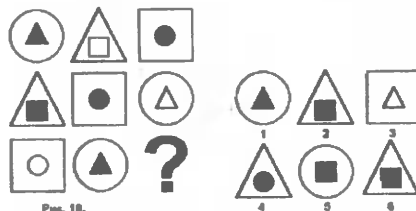


Рис. 10.

6. Поставьте в ряд или столбце квадрата столько крестиков, сколько их должно быть в соответствии с числом, написанным с внешней стороны квадрата.

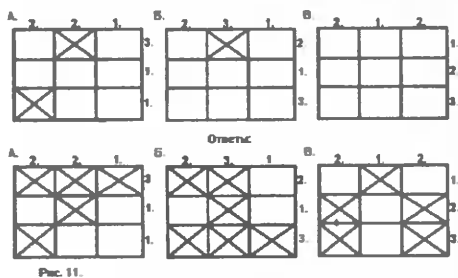


Рис. 11.

7. Коля и Саша носят фамилии Гвоздев и Шилов. Какую фамилию имеет каждый из них, если Саша с Шиловым живут в соседних домах?

8. Зина и Вера имеют фамилии Орлова и Скворцова. Какую фамилию имеет каждая из них, если известно, что Зина на 2 года моложе Орловой?

9. В соревновании по бегу Ваня, Гриша и Дима заняли первые три места. Какое место занял каждый из ребят, если Гриша занял не второе и не третье место, а Дима — не третье?

10. Три девочки — Маша, Катя и Женя — одеты в платья различных цветов — синее, желтое и белое. У Маши платье не белое, у Кати платье не белое и не желтое. Скажите, какого цвета платье у каждой из девочек.

11. Три ученицы — Тополева, Березкина и Кленова — на пришкольном участке посадили три дерева: березку, тополь и клен. Причем ни одна из них не посадила дерево той породы, от которой произошла ее фамилия. Узнайте какой породы дерево посадила каждая из девочек, если известно, что Кленова посадила не березку.

12. Три подружки — Вера, Оля и Таня — пошли в лес по ягоды. Для сбора ягод у них были корзинка, лукошко, ведерко. Известно, что Оля была не с корзинкой и не с лукошком, Вера — не с лукошком. Что с собой взяла каждая из девочек?

13. Трех котят держали девочки на руках: рыжего, черного и белого. Фамилии девочек были: Рыжова, Белова и Чернова. Ни одна из девочек не держала котенка того цвета, от которого произошла ее фамилия. Белова внимательно разглядывала черного котенка, которого держала подруга. Какого цвета котята находились на руках у каждой из девочек?

14. Сидит дитя и плачет. Прохожий спрашивает: «Что ты плачешь?» Дитя отвечает: «Есть у меня отец и мать, а я им не сын. Кто же я?» Ответьте на этот вопрос.

15. Раскрасьте маленькие флажки так, чтобы большой флажок был между желтым и зеленым, а желтый был рядом с зеленым.



Рис. 12.

16. Раскрасьте большие квадраты так, чтобы маленький квадрат был между желтым и зеленым, а черный был рядом с желтым.



Рис. 13.

17. Круг положите справа от квадрата, но слева от треугольника. Проверьте себя по образцу.



Рис. 14.

18. Знак «плюс» поставьте справа от знака «минус», но слева от знака «равно».

19. Толя веселее, чем Катя, Катя веселее, чем Алик. Кто веселее всех?

20. Волосы у Вовы светлее, чем у Пети, а волосы у Пети светлее, чем у Коли. Кто темнее всех?

21. Два мальчика играли на гитаре, а один на балалайке. На чем играл Юра, если Миша с Петей и Петя с Юрой играли на разных инструментах?

22. Определите, какого цвета линия будет на простой бумаге после проведения карандашом линии по копировальной бумаге, если:

а) бумага (некопировальная) желтого цвета, на ней сверху копировальная бумага синего цвета, карандаш зеленый;

б) бумага синяя, копия желтая, вторая копия (которую положили сверху на первую) красная, карандаш красный.

23. Через 5 лет Боре будет столько же лет, сколько Маше сейчас. Кто младше?

Помимо приведенных логических задач первоклассникам целесообразно предлагать алгоритмические этюды. Они вплотную приводят к понятию алгоритма и формализации записи. Их решение способствует формированию алгоритмического и логического мышления. Существует множество таких этюдов — от известных задач перевоза и переливания до реализованных на компьютере «Ханойской башни» и «Шахматных коней».

При подготовке алгоритмических этюдов следует продумывать методические вопросы, а именно:

- какие материалы и пособия понадобятся для этюда;
- как с наименьшей затратой времени познакомиться ребят с правилами действий;
- на какое время должно быть рассчитано решение, чтобы дети пожелали еще раз вернуться к этому этюду;
- как организовать наблюдение за



детьми, чтобы выяснить, заинтересовало ли их решение;

- какие изменения можно внести в эту задачу, чтобы повысить интерес и активность детей;
- какие выводы следует сообщить детям в заключение (назвать наиболее активных участников, недочеты и т. п.).

Приведем ряд алгоритмических задач.

1. Как, имея два сосуда емкостью 5 и 8 литров, набрать из водопроводного крана 3 литра воды?

2. Как, имея два сосуда емкостью 3 и 5 литров, набрать из водопроводного крана 7 литров воды?

Задачи на переливание, предназначенные для решения на компьютере (Переливашка, Роботландия).

3. Ведро А (полное) вмещает 3 литра, ведро Б (пустое) — 2 литра, а ведро В (тоже пустое) — 1 литр. Требуется получить в ведре А 2 литра, в ведре Б 1 литр.

4. Ведро А (полное) вмещает 6 литров воды, ведро Б (пустое) — 4 литра, а ведро В (тоже пустое) — 2 литра. Требуется получить по 2 литра воды в каждом из ведер А, Б, В.

5. Ведро А (полное) вмещает 5 литров воды, ведро Б (пустое) — 3 литра, а ведро В (тоже пустое) — 2 литра. Требуется получить за 3 хода в ведре А — 4 литра, в ведре Б — 1 литр.

6. Ведро А (полное) вмещает 4 литра, ведро Б (пустое) — 3 литра, а ведро В (тоже пустое) — 1 литр. Требуется получить за 3 хода в ведрах А и Б по 2 литра.

7. Три белые и три черные шашки расположены так, как показано на рис. 15. Требуется переставить белые шашки на место черных, а черные — на место белых с соблюдением условий:

- каждая шашка может перескочить на ближайшую клетку или через одну клетку, но не дальше;
- никакая шашка не должна возвращаться в клетку, где она уже была;
- в каждой клетке не должно быть более одной шашки.



Рис. 15.

8. В свободную клетку можно перемещать карточку, располагающуюся как в соседней клетке, так и через одну. Расставить карточки с буквами С С Т Т ... так, чтобы одинаковые буквы занимали такие же места, что и одинаковые цифры: ... 2 4 4, за три хода.

9. Буквы, расположенные слева, образуют начальное положение в задаче, а те же буквы, находящиеся справа, — их конечную, требуемую позицию. За один ход можно сделать только одно перемещение любой буквы на свободное место, остальные буквы должны оставаться на своих местах.

|       |     |     |    |
|-------|-----|-----|----|
| 1. НТ | ___ | ___ | Т  |
| В     | ___ | ___ | НВ |
| 2. КН | ___ | ___ | ПК |
| П     | ___ | ___ | Н  |
| 3. ВС | ___ | ___ | СТ |
| Т     | ___ | ___ | В  |
| 4. Б  | ___ | ___ | Ш  |
| ШФ    | ___ | ___ | ФБ |

10. «Шаги петуха».

Правила перемещения: петух все время меняет шаги, то прямо, то наискосок. Он не прыгает, а только шагает и 2 раза в одну клетку не заходит. Назвать:

- 2 шага от \*\* к -;
- 2 шага от ,, к -;
- 3 шага от . к .;
- 3 шага от оо к -;
- 4 шага от // к -;
- 4 шага от + к .

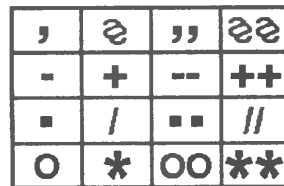


Рис. 16.

Следующий набор заданий — это алгоритмы формирования составов.

Заранее договоримся о следующих обозначениях:

- туннель — ТЛ;
- тупик — Т;
- весы — В;
- локомотив закрашивается черным цветом;
- вагоны серые и заштрихованные.

11. На железнодорожных путях (рис. 17) находятся локомотив и два вагона. В тупик Т может войти либо локомотив, либо один вагон. Машинисту необходимо кратчайшим путем (сделав наименьшее число ходов) сформировать состав, как показано на рис. 18.



Рис. 17.



Рис. 18.

12. На железнодорожных путях (рис. 19) находятся локомотив и два вагона. В тупик Т может войти либо локомотив, либо один вагон. Машинисту необходимо расположить вагоны, как показано на рис. 20.

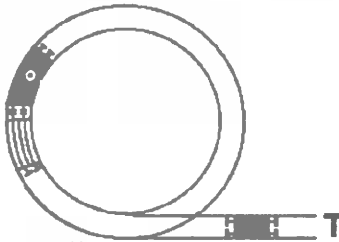


Рис. 19.

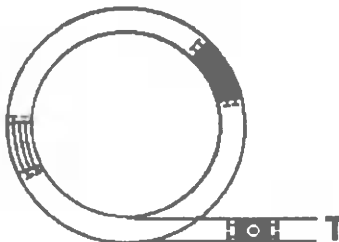


Рис. 20.

13. На железнодорожных путях (рис. 21) находятся локомотив и три вагона. Задача машиниста — перегнать локомотив и состав и расположить их на путях так, как показано на рис. 22.

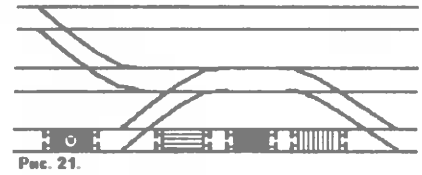


Рис. 21.

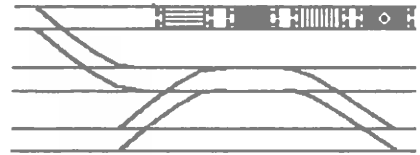


Рис. 22.

В программу курса входит много дидактических игр. Приведем их описание.

#### Игра «Отгадай предмет».

Один ученик выходит из класса. Оставшиеся задумывают какой-либо предмет, находящийся в классе. Ученик входит и начинает задавать вопросы для выяснения, какой предмет загадан. При чем вопросы нужно задавать так, чтобы на них можно было ответить либо «да», либо «нет».

Цель данной игры: показать учащимся, что информацию можно получить, задавая вопросы и получая на них ответы.

#### Игра «Телефон».

Эта игра напоминает игру «Испорченный телефон», только цель здесь другая: постараться сохранить информацию, сообщенную учителем.

Цель данной игры: показать возможный способ передачи информации и сделать вывод о том, что чем длиннее канал передачи информации, тем более вероятно ее искажение.

#### Игры «Шифровка» и «Разведчик».

Учащиеся делятся на две команды. Одна команда придумывает шифр и с его помощью зашифровывает некоторое сообщение. Другая команда по исходному и зашифрованному сообщению отгадывает шифр. Затем команды меняются ролями.

Цель данной игры: научить кодировать и декодировать информацию, показать разнообразие способов кодирования информации.

#### Игра «Аукцион».

Учащиеся делятся на две команды, между которыми проводится соревнование, заключающееся в том, чтобы за

5 минут сообщить как можно больше интересных фактов, связанных с применением компьютеров. Атрибутом аукциона является деревянный молоток в руках ведущего. При первом ударе молотка ведущий спрашивает: «Кто еще добавит новые факты?» Если команда молчит, делает еще удар и спрашивает: «Не вспомнил ли кто еще?» Если ответа не последует, ведущий делает третий удар и объявляет число высказанных фактов.

#### Игра «Расположи предмет».

Цель данной игры: сформировать у детей умение ориентироваться на листе бумаги (в клетку), находить заданную точку по ориентирам, понимать выражения: «отступить вправо, влево, вверх, вниз», «верхняя строчка», «нижняя строчка».

Детям раздаются специально расчерченные листы в крупную клетку 1x1 см. Далее их просят найти нужную клетку и расположить в ней некоторый рисунок. Например, отсчитать от нижней строчки 3-й ряд вверх, найти в нем 10-ю строчку слева, в этой клетке нарисовать яблоко и т. д.

Следующий этап — выполнение задания под диктовку. От указанной точки надо двигаться в том направлении, которое указывает учитель. Ученик, выполняющий рисунок, не знает, что у него получится (в этом есть элемент неожиданности). Когда изображение готово, ученик называет, что он нарисовал.

Данная игра хорошо подготавливает учеников к работе в графическом редакторе.

#### Логическая игра «Какого цвета твоя шапка?».

Для проведения игры необходимо из бумаги изготовить 4 шапочки: 2 шапочки белого цвета и 2 — синие.

В один ряд в затылок друг другу садятся 3 ученика, они не оглядываются. Начиная с заднего (третьего) ученика им на головы надевают шапочки: третьему — синюю, второму и первому — белые. Остальные шапочки прячут. Каждый из этих ребят не должен знать, какого цвета шапочка на его голове. Последний из сидящих ребят видит только цвета шапочек первых двух, средний — цвет шапочки только переднего ученика. Отгадывать должны в таком порядке: сначала последний из ребят (задний), затем второй, а потом первый.

**Объяснение:** так как третий ученик видит, что все белые шапочки на впереди

сидящих ребятах (их всего две), то он сразу говорит, что на его голове синяя шапочка. Наблюдающие должны подтвердить это. Услышав, что третий ученик угадал цвет своей шапочки, второй, видя на переднем ученике белую шапочку, догадывается, что на нем тоже белая шапочка. Присутствующие ребята и его ответ подтверждают. Первый ученик по ответам ребят догадывается, что и на нем белая шапочка.

#### Логическая игра «Угадай название цветка на карточке».

Для игры берутся 2 карточки с изображением маков и 2 карточки с изображением васильков.

В один ряд в затылок друг другу садятся 3 ученика. Начиная с третьего на спины учеников навешивают карточки так, чтобы у первого была карточка с изображением мака, у второго — с изображением василька, а у третьего — любая из оставшихся карточек.

Ответы даются начиная с последнего, третьего ученика, который видит две карточки.

**Объяснение.** Третий ученик по двум увиденным карточкам может только сказать, что он не знает, какой цветок изображен на его карточке. Учитывая ответ третьего ученика и видя карточку у первого ученика, второй делает вывод, что на его карточке изображен василек. Услышав два ответа, первый ученик догадывается, что на его карточке изображен мак.

После всех высказываний ученики объясняют ответы.

#### Игра «Вычислительная машина».

Каждый ученик, сидящий слева за партой, будет выполнять роль ЭВМ — считать быстро и правильно, а другой, сидящий справа, — роль контролера. Учитель заранее чертит схему на доске и предлагает ученикам посмотреть на нее.

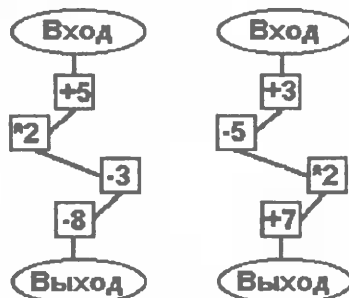


Рис. 23.

Ученики-контролеры запускают машину, кладут на вход некоторое число. Тогда ученики, выполняющие роль ЭВМ, производят все вычисления, указанные на схеме, и кладут на выход нужные карточки с цифрами, обозначающими число (ответ составного примера). Затем они показывают после проверки контролеров это число учителю.

#### Задачи с неполным условием.

Учитель сообщает ученикам условие задачи и вопрос к ней. Ученики должны задавать вопросы, на которые можно было бы ответить только «да» или «нет», при этом ученики должны внимательно слушать ответы не только на свои вопросы, но и на вопросы товарищей. В результате, собрав необходимую информацию, они дают ответ на вопрос задачи.

#### Задача 1.

Один ученик, купаясь в речке, нашел необычный сосуд, после чего его жизнь резко изменилась. Что произошло? (Встретились пионер Волька и старик Хоттабыч.)

#### Задача 2.

Один человек зашел в магазин, вынул в кассе 1 рубль в третий отдел, внимательно посмотрел на чек, затем выбросил его и вышел из магазина. С какой

целью он выбивал чек? (Он хотел узнать сегодняшнее число.)

#### Задача 3.

Одна девочка получила в подарок цветок и поэтому смогла побывать на Северном полюсе. Кто была эта девочка? (Девочка Женя из сказки «Цветик-семицветик».)

#### Задача 4.

Два человека подошли к реке. У пустого берега стояла лодка, в которой мог поместиться только один человек. Оба без всякой помощи переправились на этой лодке через реку и продолжили свой путь. Как они это сделали? (Двое подошли к разным берегам реки.)

#### Задача 5.

Встретились два друга детства, не видевшиеся со школьных лет и ничего не знавшие друг о друге. Между ними состоялся разговор:

— Сколько лет я тебя не видел и не получал никаких вестей!

— А у меня уже дочь.

— Как же ее зовут?

— Да так же, как и ее мать.

— А сколько лет твоей Ниночке?

Как один из собеседников узнал имя дочери другого?

(Вторым собеседником была женщина, которую звали Нина.)

### *Уважаемые читатели!*

Напоминаем вам, что 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

**А. В. Горячев,**

*руководитель авторского коллектива курса «Информатика в играх и задачах»*

## «ИНФОРМАТИКА В ИГРАХ И ЗАДАЧАХ»

Как многие науки, информатику можно условно разделить на теоретическую и прикладную. Теоретическая информатика — математическая дисциплина. Она использует методы математики для построения и изучения моделей обработки, передачи и использования информации, создает тот фундамент, на котором строится все здание информатики. Заимствуя многие модели из дискретной математики, теоретическая информатика, как правило, наполняет их конкретным содержанием, связанным со спецификой информации.

Курс «Информатика в играх и задачах» для начальной школы представляет собой авторское переложение основных дисциплин теоретической информатики для начальной школы. Актуальность нового курса определяется необходимостью подготовки мышления учеников к восприятию ими в будущем современных информационных технологий. Работая в сотрудничестве со специалистами Московского экономико-статистического института, авторы наблюдали, как преподаватели Центра довузовского образования МЭСИ проводят подготовительные занятия в старших классах некоторых московских школ, пытаясь в дальнейшем облегчить освоение студентами современных технологий. К сожалению, усиливая, направляемые на интенсификацию подготовки по информатике, наталкиваются на неготовность большинства учеников легко и быстро воспринимать новые идеи и технологии, на определенную неразвитость логического мышления. Психологи утверждают, что основные логические структуры мышления формируются в возрасте 5—11 лет, а запоздалое формирование этих структур протекает с большими трудностями и часто остается незавершенным.

Обучение теоретической информатике позволяет ответить на вопрос: каким образом максимально подготовить учеников к грамотному использованию современных технологий, избегая в то же время жесткой профессиональной ориентации? Имея дело с различными сторонами действительности, окружающей челове-

ка, информационные технологии обладают рядом методов, подходов, приемов, позволяющих анализировать и моделировать мир вокруг нас для решения возникающих задач. Это и алгоритмический подход, и объектно-ориентированный анализ с последующим объектно-ориентированным проектированием, и снабжение компьютеров базами знаний для перекладывания на них рутинного умственного труда, освобождающего человека для творческой деятельности. Овладевая этими приемами, человек учится мыслить логично, стройно, системно. Полезно ли это всем? Вне всякого сомнения.

Основная цель курса, определенная узко, — это пропедевтика информатики, а определенная более широко — это подготовка к решению разнообразнейших задач в области информационных технологий, и она может быть рассмотрена более подробно — как совокупность более конкретизированных целей:

1. Развитие у школьников устойчивых навыков решения задач с применением таких подходов к решению, которые наиболее типичны и распространены в областях деятельности, связанных с информационными технологиями:

- использование формальной логики при решении задач (построение выводов путем применения к известным утверждениям логических операций «если то», «и», «или», «не» и их комбинаций — «если ... и ... то ...»);
- алгоритмический подход к решению задач (умение планировать последовательность действий для достижения какой-либо цели, а также решать широкий класс задач, для которых ответом является не число или утверждение, а описание последовательности действий, например задача про волка, козу и капусту);
- системный подход (рассмотрение сложных объектов и явлений в виде набора более простых составных частей, каждая из которых выполняет свою роль для функционирования объекта в целом: рассмотрение влияния изменений в одной составной

части на поведение всей системы — «как вы думаете, что будет, если...»);

- объектно-ориентированный подход (умение объединять отдельные предметы в группу с общим названием, выделять общие признаки предметов этой группы и действия, выполняемые над этими предметами: умение описывать предмет по принципу «из чего состоит и что делает или можно с ним делать»).

2. Расширение кругозора в таких областях знаний, тесно связанных с информатикой (прежде всего в математике), с которыми школа недостаточно знакомит или не знакомит вовсе (по практическому опыту преподавателей МЭСИ в школах Москвы). В начальной школе речь идет о знакомстве с графами, комбинаторными задачками, логическими играми с выигрышной стратегией («начинают и выигрывают») и некоторыми другими.

Несмотря на ознакомительный подход к данным дисциплинам, по каждой из них предполагается обучение решению простейших типовых задач, включаемых в контрольный материал, т. е. акцент ставится на умении приложения даже самых скромных знаний.

3. Создание у учеников навыков решения логических задач и ознакомление с общими приемами решения задач — «как решать задачу, которую раньше не решали» (поиск закономерностей, рассуждения по аналогии, по индукции, правдоподобные догадки, знакомство с теорией решения изобретательских задач и др.).

Первый год обучения теоретической информатике рассчитан на 7-летних детей, что соответствует I классу по программе 1—3 и II классу по программе 1—4. В течение этого года проводится подготовка к предстоящим в дальнейшем занятиям, развивается логическое мышление детей и сообразительность. При проведении занятий широко применяются занимательные и игровые формы обучения. Как правило, различные темы и формы подачи учебного материала активно чередуются в течение одного урока.

Дальнейшее обучение логической информатике проводится по нескольким направлениям. Близкие направления со-

браны в 4 группы (по числу учебных четвертей). Каждая учебная четверть заканчивается контрольной работой. Таким образом ученики каждую четверть продолжают изучение темы этой же четверти прошлого года.

I четверть — алгоритмы;

II четверть — системы и объекты;

III четверть — дискретная математика и искусственный интеллект;

IV четверть — приемы решения задач и стратегии в играх.

Кроме того, задачи по каждой из тем могут быть включены в любые уроки в любой четверти в качестве разминки. Занятия проходят один раз в неделю в обычном классе без компьютеров.

Следует отметить, что недостаточно придумать и провести интересный урок, надо описать его так, чтобы уроки, проводимые в соответствии с этим описанием, тоже были интересными. Именно поэтому особое внимание авторы разработки уделяли технологии тиражирования уроков, что привело к появлению подробных поурочных планов. В этих описаниях уроков можно прочитать не только о последовательности изложения материала, но и о вероятном реагировании детей во время игр и выполнения заданий. Методический материал создавался с таким расчетом, чтобы, применяя его, учитель, не имеющий достаточного опыта, мог бы провести интересный урок, а опытный учитель, зная о целях урока, — творчески внести в него какие-либо изменения.

В предлагаемый школам комплект для обучения «Информатика в играх и задачах» входят:

- учебный материал в виде тетрадок-раскрасок (одна тетрадка на одну четверть);
- контрольные работы (по два варианта к каждой тетрадке);
- методические пособия в виде подробных поурочных планов.

Для того чтобы избежать возможности разукрупнения набора в пути от издателя к учителю, в одном комплекте объединен необходимый материал на 25 человек на один год ( $4 \times 25 = 100$  тетрадок, 4 методических пособия,  $4 \times 25 \times 2 = 200$  контрольных работ).

Комплекты можно заказать в Центре довузовского образования Московского экономико-статистического института по адресу:  
119501, Москва, Нежинская ул., д. 7, МЭСИ, ДОВУЗ,  
или по факсу: 442-65-58, ДОВУЗ.

Контактные телефоны в Москве: 442-63-44, 442-64-43.

# ЗАДАЧИ

**Ю. С. Варакин,**

*учитель средней школы № 550, Москва*

## **ПРОЕКТЫ-ЗАДАНИЯ НА ЯЗЫКЕ ЛОГО ДЛЯ УЧАЩИХСЯ СТАРШИХ КЛАССОВ\***

### **Задание 7**

#### **Цели**

1. Составление и отладка программ, использующих циклические алгоритмы и параметры со случайными величинами.
2. Достижение определенного эстетического эффекта графической композиции, полученной при выполнении программы. Данную программу можно использовать, например, при проектировании расцветок тканей.

#### **Формулировка**

Составить программу для изображения абстрактного орнамента, который может быть использован при проектировании рисунка ткани.

Орнамент должен представлять собой композицию из цветных фигур (например, звездочек) случайной величины, размещенных случайным образом по цветному фону случайной расцветки.

#### **Рекомендации к выполнению**

Программу для выполнения задания можно составить из трех процедур: процедуры случайного движения черепашки по экрану дисплея, процедуры изображения геометрической фигуры и процедуры окраски каждой фигуры графической композиции.

Процедура случайного движения черепашки может представлять собой многократное повторение движения черепашки вперед на случайную величину и поворот ее на случайный угол. После каждой команды вперед черепашка выполняет процедуру, изображающую некоторую геометрическую фигуру, имеющую случайный размер. Выполнение процедуры случайного движения черепашки лучше осуществлять так, чтобы за ней не оставалось следа, т. е. перо черепашки должно быть поднято.

Процедура, изображающая геометрическую фигуру случайного размера, имеет параметр — размер, выбираемый случайным образом.

В качестве процедуры изображения геометрической фигуры может быть выбрана процедура изображения треугольника, квадрата, многолучевой звезды и т. д.

После выполнения процедуры, изображающей геометрическую фигуру, должна быть выполнена процедура закраски фигуры цветом, выбранным случайным образом. Процедура закраски должна предусматривать продвижение черепашки внутрь нарисованной фигуры и выполнение команды закраски случайно выбранным цветом (при этом перо черепашки должно быть опущено). После выполнения команды закраски черепашку желательно вернуть на исходную позицию, тогда процедура закраски будет универсальной и может применяться в других заданиях.

---

\* В данной статье мы продолжаем публикацию задач по Лого, начатую в № 1, 3, 1995. — *Примеч. ред.*

### Используемые примитивы языка Лого

|        |                |
|--------|----------------|
| fd     | — вперед       |
| bk     | — назад        |
| rt     | — направо      |
| lt     | — налево       |
| repeat | — повтори      |
| pd     | — перо опусти  |
| pu     | — перо подними |
| random | — случайно     |

### Программа

```

to орнамент
ht cg repeat 100[pu fd 25+random 150 звезда rt random 360]
end

to звезда
name random 40+5 "n setc 1 pd repeat 5[fd :n lt 360/5 fd :n rt(360/5*2)]
закраска
end

to закраска
setc random 115 pu lt 45 fd 2 pd fill bk 2 rt 45
end

```

### Задание 8

#### Цели

1. Составление и отладка программ, использующих циклические алгоритмы с изменяющимся в процессе цикла параметром.
2. Получение декоративной графической композиции.

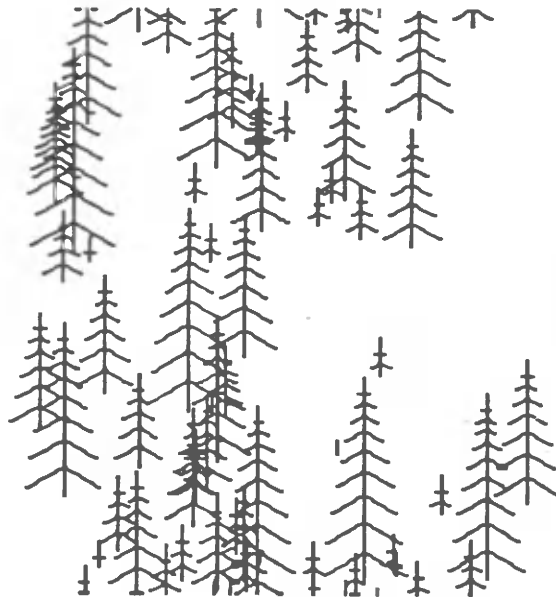


Рис. 1



### Формулировка

Составить программу, в результате работы которой на экране будет получена графическая композиция с изображением елового леса (рис. 1).

Фигурки елей могут быть окрашены зеленым или произвольным цветом.

### Рекомендации к выполнению

Программу для выполнения предложенной графической композиции можно представить в виде двух процедур: процедуры выполнения рисунка ели произвольной величины и процедуры расстановки елей по экрану.

Процедуру выполнения рисунка ели можно представить как повторение рисунка одного яруса ветвей ели. При этом, если изображение ели начинать с нижнего яруса, размеры элементов яруса должны уменьшаться при повторении рисунка каждого следующего яруса. Если в процедуре выбрать параметр  $n$ , означающий число ярусов ели, то отдельные элементы яруса можно выразить через выбранный параметр. Этот параметр при каждом повторении рисунка яруса ели должен уменьшаться, к примеру, на единицу. В результате рисования определенного числа ярусов ели будет получен рисунок с уменьшающейся величиной отдельных ярусов. В процедуре также необходимо установить цвет (зеленый или какой-то другой — по желанию) и такой курс черепашки, чтобы рисунок каждой ели был вертикальным.

Процедура расстановки елей по экрану дисплея представляет собой случайное движение черепашки. После продвижения на случайную величину выполняется предыдущая процедура, после чего черепашка поворачивается на случайный угол и процедура движения черепашки повторяется.

### Используемые примитивы языка Лого

|        |                            |
|--------|----------------------------|
| bk     | — назад                    |
| fd     | — вперед                   |
| rt     | — направо                  |
| pu     | — перо поднять             |
| pd     | — перо опустить            |
| seth   | — установи курс            |
| setc   | — установи цвет            |
| repeat | — повтори                  |
| random | — случайно                 |
| name   | — имя локальной переменной |

### Программа

```
to ель n
  seth 0 pd setc random 13
  repeat :n[fd :n+3 rt 120 fd :n*2 bk :n*2 rt 120 fd :n*2 bk :n*2 rt 120 name :n-1 "n] fd 5
end
```

```
to лес
  ht cg
  repeat 200[pu fd 5+random 80 ель random 10 rt random 180]
end
```

## Задание 9

### Цели

1. Составление и отладка программ, использующих локальные и глобальные переменные.
2. Получение программы для описания достаточно простой компьютерной игры.

### Формулировка

Составить программу компьютерной игры «Кто надует самый большой воздушный шар?».

Смысл игры состоит в следующем. Первый игрок надувает шар за несколько попыток, в каждой попытке можно увеличить его на случайную величину от 3 до

11 условных единиц. Если в результате нескольких попыток шар достигает величины, превышающей число 21, то он лопается и игрок проигрывает, что отражается соответствующим сообщением на экране. Первый игрок может прекратить игру, получив шар достаточной величины, тогда цель второго игрока — получить шар большего размера, но так, чтобы он не лопнул (рис. 2).



Рис. 2

### Рекомендации к выполнению

Программа для описания предложенной компьютерной игры включает три процедуры.

В первой процедуре можно выполнить очистку экрана от предыдущей графики и определить две глобальные переменные для накопления суммы значений увеличения шаров в каждой попытке для первого и второго игрока. Очевидно, что первоначальное значение этих переменных должно быть нулевым.

Во второй процедуре учитывается увеличение размера шара  $n$  для одной попытки. В процедуре определяется локальная переменная  $n$ , величина которой случайна, и глобальная переменная  $m$ , которая учитывает увеличение размера шара при каждой попытке, при этом значение  $m$  выводится на экран. Если значение  $m$  больше 21, то изображение шара на экране пропадает, появляется сообщение «Вы проиграли» и выполнение процедуры прекращается. Если значение  $m$  меньше 21, то слева от центра экрана изображается шар, объем которого пропорционален значению  $m$ . Выполнив данную процедуру несколько раз, первый игрок либо получает достаточную величину шара, либо проигрывает, если величина шара превышает заданное значение.

Для того чтобы второй игрок имел возможность получить изображение своего шара, нужно продублировать первую процедуру для переменной  $m1$  и рисунок шара выполнять правее от центра экрана.

Рисунки шаров, образуемых двумя последними процедурами, можно закрасить определенным цветом, для чего необходимо сформировать отдельную процедуру закраски. При этом нужно учесть, что для закрасивания всего контура шара черепашка должна попадать в незакрашенную часть шара. Это можно сделать, связав величину продвижения черепашки внутрь контура с величинами  $m$  или  $m1$ , определяемых в процедурах, описанных выше.

### Используемые примитивы языка Лого

|      |                    |
|------|--------------------|
| nt   | — спрячь черепашку |
| fd   | — вперед           |
| bk   | — назад            |
| rt   | — направо          |
| lt   | — налево           |
| fill | — закрась          |
| pu   | — перо поднять     |

|        |                                       |
|--------|---------------------------------------|
| pd     | — перо опустить                       |
| setc   | — установить цвет                     |
| cg     | — сотри графику                       |
| ct     | — сотри текст                         |
| name   | — имя локальной переменной            |
| make   | — имя глобальной переменной           |
| random | — случайный                           |
| repeat | — повторить                           |
| pr     | — напечатать                          |
| if     | — если                                |
| stop   | — стоп                                |
| setx   | — перевести черепашку по координате x |

### Программа

```
to игра
  setc 5 ht cg ct make "m 0
  make "m1 0
end
```

```
to игрок1
  name 3+random 9 "n make "m :m+:n pr :m
  if :m21 [cg pr "proigrali stop]
  pu cg setx -40 pd repeat 12[fd :m rt 30]закраска :m
end
```

```
to игрок2
  name 3+random 9 "n make "m1 :m1+:n pr :m1
  if :m121 [cg pr "proigrali stop]
  pu home setx 50 pd repeat 12[fd :m1 rt 30]закраска :m1
end
```

```
to закраска k
  pu setc 5 rt 90 fd :k*3.5 pd fill pu bk :k*3.5 lt 90 pu rt 90 fd :k
end
```

## Предлагаем пользователям «Агат-7/9»

*Новую ОС и программное обеспечение.*

*Цифровой синтезатор SOVOX и ПО для него.*

Вышлем каталог, демо-диск.

Адрес: 630117, Новосибирск, а/я 427.

Л. Ю. Белова, В. А. Кузнецова, Ю. А. Белов, С. В. Епимахов,  
г. Ярославль

## ЗАДАЧА ИЗОБРАЖЕНИЯ СЕЧЕНИЙ МНОГОГРАННИКОВ

Цель статьи — кратко познакомить с одной из разработок, которая создана в рамках семинара по машинной графике, действующего в Ярославском университете. В настоящее время система проходит опытную эксплуатацию в «пилотной» школе № 76 г. Ярославля. Речь пойдет о диалоговой программной системе, обучающей общему методу построения на чертеже плоских сечений многогранников — методу следа. В качестве многогранников могут выбираться усеченные или неусеченные пирамиды и обобщенные призмы. Обобщенными призмами здесь названы многогранники с параллельными боковыми ребрами и, возможно, с непараллельными основаниями.

Кратко напомним метод. Как известно, чтобы построить сечение, имея изображение многогранника, надо найти или точки пересечения ребер многогранника с секущей плоскостью или отрезки, по которым грани многогранника пересекают плоскость. Отметим, что каждая точка пространства на чертеже представляется двумя точками — своим изображением и проекцией на плоскость основания многогранника (см. рис. 1). При этом в случае призм используется парал-

лельное проектирование, определяемое направлением боковых ребер, а в случае рассмотрения пирамид — центральное проектирование из вершины пирамиды. С точностью до вида проектирования решение задач для обоих типов многогранников фактически проводится одинаково. Поэтому далее для краткости ограничимся примером призмы.

В методе следа основной этап построений — нахождение пересечения секущей плоскости с боковыми ребрами (или их продолжениями).

Далее при необходимости находятся дополнительные точки — пересечения плоскости с ребрами верхнего и нижнего оснований, после чего все вершины многоугольника сечения получены.

На предварительном этапе находится след — прямая пересечения секущей плоскости с основанием призмы.

Секущая плоскость обычно задается тремя точками, которые представляются на чертеже своими изображениями и проекциями на плоскость основания. На рис. 1 это, например, точки А, В, С и их проекции А1, В1, С1. В простейших случаях точки секущей плоскости располагаются на боковых ребрах, и тогда их

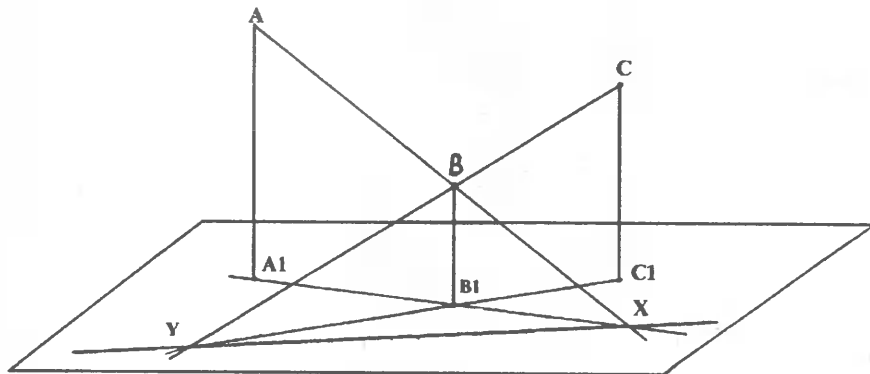


Рис. 1

проекциями являются соответствующие вершины нижнего основания.

Способ построения следа показан на рис. 1. Ученик должен провести прямые  $A;B$  и  $A1;B1$  — их пересечение даст одну точку следа —  $Y$ , аналогично находится точка  $X$ .

После того как след построен, поочередно строятся точки пересечения плоскости  $ABC$  с боковыми ребрами — см. рис. 2, где показано построение пересечения  $ABC$  с ребром  $D;D1$ . Сначала находится вспомогательная точка  $D2$  — пересечение следа с  $B1;D1$ . Затем находится точка  $D3$  на ребре. Вспомогательные линии автоматически гасятся после получения необходимых построений, чтобы не загромождать чертеж. Конечно, приводимые построения требуют обоснования, но они несложны и представляются читателю.

Построение точек пересечения секущей плоскости с верхним и нижним основаниями проводится в зависимости от расположения точек, полученных на боковых ребрах или их продолжениях.

#### О составе системы.

Комплекс включает в себя три исполняемых файла с программами, предназначенными ученику — `teacher`, учителю — `gred` и промежуточной — `gredcom` — с техническими функциями.

В ученической программе на экране задается чертеж с изображением соответствующего многогранника и трех точек плоскости сечения, дается возможность проводить любую нужную прямую, указав имена двух точек на ней, и находить

точки пересечения прямых. Попытки действий, не требующихся в методе, отмечаются и суммируются как ошибки соответствующего вида. Учитывается и количество обращений к помощи, где имеется краткое описание метода и эталонная демонстрационная процедура построения сечения. Перед выходом программа дает заключение о правильности построения сечения и итоговое количество ошибок.

Ученическая программа может работать с различными чертежами, информация о которых хранится в специальных файлах данных. Создание этих файлов проводится с помощью программы `gred` — специализированного графического редактора. С его помощью учитель может накапливать контрольные задания для ученической программы. При этом для составления чертежа приходится проделать фактически все действия, с которыми потом столкнется ученик, строя сечение. Кроме того, при построении некоторых точек надо указывать дополнительную информацию, благодаря чему эти точки будут в дальнейшем правильно восприниматься программой `teacher`. Атрибуты у точек могут быть следующие: точка следа, основания, проекция другой точки, вершина пирамиды, точка секущей плоскости. Атрибуты строящейся точки предлагаются в соответствующем меню и могут быть использованы по желанию.

После создания нового файла геометрических данных или редактирования имевшегося чертежа необходимо

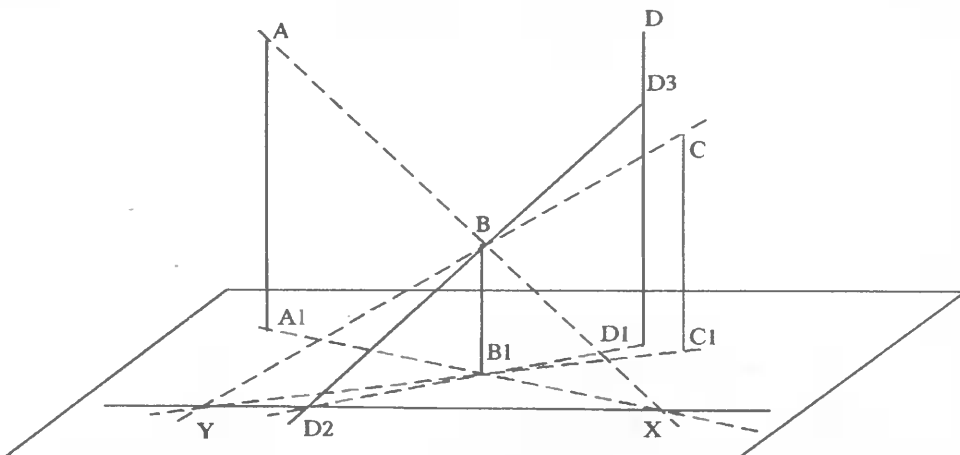


Рис. 2

прямо из программы `gred` запустить контрольную программу `gredcom`, работающую в автоматическом режиме. Ее задача — проверка правильности расстановки атрибутов и некоторое преобразование геометрической базы данных, связанной с чертежом. Программа `gredcom` запускается из основного меню редактора `gred`, выдает соответствующую диагностику о построенном чертеже и возвращает управление в редактор.

В систему входят также руководства для ученика и для учителя с описанием метода и пользовательскими инструкциями. Руководства представлены в виде текстовых файлов и нескольких рисунков или в виде брошюры.

Программный комплекс реализован в `TURBO-C 2.0` и работает на ПЭВМ, совместимых с `IBM PC/XT,AT`. Объем исходных текстов: `gred` — 740 строк, исполняемый файл — 33 Кб, `teacher` — 400 строк, 26 Кб, `gredcom` — 350 строк, 14 Кб.

Отметим, что имеется более простая версия обучающей системы для тех же классов многогранников, реализованная на `ЯМАХЕ` на Паскале. Основное отличие от описанного комплекса состоит в том, что в этом варианте файлы с чертежами надо готовить вручную: задавать координаты нужных точек на чертеже и т. п. Впрочем, некоторый набор

файлов уже накоплен, и его можно скопировать.

Конечно, можно заметить, что такая довольно большая система служит для обучения методу решения задач, которым в учебнике [2] уделено всего две страницы. Однако, на наш взгляд, из-за малого количества задач, развивающих пространственное воображение, школьники (а иногда и студенты) сталкиваются с различными трудностями при изучении геометрии, анализа и конструкторских дисциплин, а предлагаемая система способствует ликвидации этого пробела. Хотелось бы надеяться также, что общее толкование метода, реализованное в системе, пригодится для школ и классов с углубленным изучением математики, а также при изучении соответствующей темы начертательной геометрии в вузах. Небольшая модификация системы позволяет приспособить ее к обучению другому методу построения сечений, например методу внутреннего проецирования, а также к другим близким задачам, например о пересечении многогранника и прямой или двух многогранников, также рассматриваемым в курсах начертательной геометрии и инженерной графики.

Наш адрес: 150000, г. Ярославль, ул. Советская, д. 14, ЯрГУ, кафедра теоретической информатики, Белову Юрию Анатольевичу.

Телефоны: 11-02-66, 21-01-10.

## Литература

1. Начертательная геометрия/Под ред. Н. Н. Крылова. М.: Высшая школа, 1990.

2. Погорелов А. В. Геометрия VI—X: Учебное пособие для VI—X классов средней школы. М.: Просвещение, 1988.

\* \* \*

## ОТВЕТЫ НА КРОССВОРД, ОПУБЛИКОВАННЫЙ НА СТР. 46

*По горизонтали:* 5. Полетаев. 6. Проверка. 8. Генератор. 11. Поиск. 14. Длина. 15. Алфавит. 16. Сдвиг. 17. Минск. 20. Зарплата. 22. Отбор. 23. Биакс. 26. Состояние. 27. Парадокс. 28. Итератор. *По вертикали:* 1. Контроль. 2. Пакет. 3. Довод. 4. Указание. 7. Фраза. 9. Псевдокод. 10. Следствие. 12. Плагиат. 13. Диаметр. 18. Интервал. 19. Маккалок. 21. Логос. 24. Робот. 25. Винер.

# APPLE ДЛЯ ОБРАЗОВАНИЯ

**Г. С. Николайчук,**

*учитель-методист, учебно-методический центр информатизации образования  
Красносельского района, Санкт-Петербурга*

**Ю. Ф. Титова,**

*учитель, учебно-методический центр информатизации образования Красносельского  
района, Санкт-Петербурга*

## ГРАФИЧЕСКИЙ РЕДАКТОР KIDPIX НА УРОКАХ ИНФОРМАТИКИ

### Введение

В настоящее время компьютерами Macintosh фирмы Apple оснащаются все больше школ России. Для работы с детьми эти компьютеры укомплектованы достаточно разнообразным программным обеспечением. Когда в 1993 г. наш учебно-методический центр приобрел класс компьютеров Macintosh LC-II, первая задача состояла в том, чтобы найти методический подход к применению этих великолепных средств на уроках информатики. В частности, осваивая графический редактор KidPix, имеющийся практически у каждого пользователя Macintosh, мы совершенно по-новому взглянули на проблему преподавания темы «Графические средства ЭВМ» в курсе информатики.

До сих пор графический редактор рассматривался в курсе информатики как некий экзотический набор юного художника. Его изучению отводилось 2—3 часа практического времени, за которое учащиеся успевали вдоволь нарисоваться и потерять к нему всякий интерес.

На наш взгляд, для работы с графическими редакторами нужно выделять учебные часы в течение 2—3 лет обучения, с каждым годом усложняя задания и расширяя рамки применимости.

Нами разработан цикл лабораторных и творческих работ по изучению KidPix. Часть работ посвящена освоению инструментальных средств редактора, другая же обширная часть — комплекс-

ному применению инструментов и команд для решения разнообразных обучающих задач. Здесь находят свое отражение межпредметные связи с такими общеобразовательными дисциплинами, как геометрия, стереометрия, черчение, география, русский язык. Общая направленность работ — творческое развитие ребенка.

Работы рассчитаны на разный уровень подготовленности учащихся.

На начальном этапе каждое задание сопровождается подробным алгоритмом (вплоть до указания, каким инструментом и как выполнить построение). В дальнейшем алгоритмы становятся более краткими. Учащимся, хорошо овладевшим средствами KidPix и умело применяющим их для достижения цели, задания выдаются без сопровождающих алгоритмов. На примере таких заданий совсем невычислительного характера мы стараемся развить у ребят навыки алгоритмического подхода к решению задач, что тесно перекликается с темой «Алгоритмизация» курса информатики.

В отдельных лабораторных работах используются заранее подготовленные в том же редакторе и защищенные от записи файлы-заготовки. Эти файлы открываются перед началом работы, и на их основе выполняются задания.

Разнообразие предлагаемых работ, творческая фантазия учеников, возможности графического редактора KidPix дают великолепные результаты при обучении не только информатике, но и другим предметам.

Конечно, мы понимаем, что KidPix во многом превосходит графические редакторы, с которыми приходится работать на уроках нашим коллегам — учителям информатики, особенно имеющим дело с отечественной техникой. KidPix, сделанный с большой любовью к детям, имеет множество штампов, шаблонов для раскрашивания, которые дети могут вставить в свой рисунок, спецэффектов, творящих чудеса, волшебную кисть и богатейшую палитру. Добавьте к этому мышь всего с одной кнопкой, послушно выполняющую все команды ребенка...

Но все же осмелимся предположить, что цель, поставленная в наших работах, может быть достигнута и с помощью других, менее совершенных графических средств.

Неспешное и полноценное освоение графического редактора возможно в случае, если информатика вводится в учебную программу с V—VI классов, поэтому наш методический материал ориентирован в основном на преподавание в V—VIII классах. Если же предмет ребята начали изучать только в X классе, то учебная программа не позволяет в полной мере узнать возможности редактора, разве только в сокращенном ускоренном варианте.

Данная статья освещает четыре аспекта работы с графическим редактором:

- методику освоения инструментальных средств;
- алгоритмический подход к выполнению работы;
- межпредметные связи и разнообразие задач;
- творческую направленность заданий.

### Методика освоения графического редактора KidPix

Графический редактор KidPix является универсальным прикладным средством для составления на ПК графических композиций, чертежей. Он включает в себя обширный набор инструментов (режимов) и команд. Дружественный пользовательский интерфейс представляет собой совокупность меню: графических в виде характерных пиктограмм и текстовых, назначение которых становится сразу понятно даже начинающему пользователю.

Первая лабораторная работа посвя-

щается знакомству с окном графического редактора KidPix. Ребята должны усвоить названия и назначение основных меню и других объектов, представленных в окне.

Основные меню редактора: МЕНЮ-СТРОКА, меню ИНСТРУМЕНТЫ, ПАЛИТРА.

МЕНЮ-СТРОКА представляет собой заголовки четырех скрытых меню команд: ФАЙЛ, РЕДАКТОР, СЕРВИС, ДРУЖОК. Эти меню позволяют выполнить любую доступную в данный момент команду (доступные команды выделены).

Меню ИНСТРУМЕНТЫ содержит пиктограммы двенадцати инструментов редактора: КАРАНДАШ, ЛИНИЯ, ПРЯМОУГОЛЬНИК, КРУГ, КИСТЬ, СПЕЦЭФФЕКТЫ, РЕЗИНКА, КРАСКА, СИМВОЛЫ, ШТАМП, ПЕРЕМЕЩЕНИЕ, ОТМЕНА. Каждый инструмент имеет одно или два дополнительных меню.

Меню ПАЛИТРА содержит 32-цветную гамму и окно ТЕКУЩИЙ ЦВЕТ.

Учащиеся знакомятся с понятием меню и его видами: графическим, текстовым, скрытым, дополнительным; с понятиями пиктограммы; доступной и недоступной команды; осваивают навыки запуска программы и выхода из нее, переключения сообщений на русский язык, выбора инструмента, цвета, дополнительного режима инструмента.

Освоение редактора начинается с инструмента РЕЗИНКА и команды ОТМЕНА.

Инструмент РЕЗИНКА позволяет уничтожать рисунок полностью или частично. Дополнительное меню содержит 14 окон с видами резинок. Среди них — резинки частичного стирания различной формы и размеров и резинки полного стирания, которые позволяют уничтожить рисунок, взорвав его бомбой, сжав его в «черную дыру», раздвинув шторки экрана... Такое обилие вариантов может кому-то показаться излишним, но, по нашему мнению, это вносит приятное разнообразие в работу и вырабатывает у ребят вкус к хорошим программным продуктам.

Команда ОТМЕНА позволяет отменить последнее выполненное действие, как бы вернуться на шаг назад: стереть последнюю нарисованную линию, фигуру, штамп; отменить последнее стирание резинкой; отменить перенос части рисунка и пр. Поскольку команда ОТМЕНА очень часто используется, то ее



пиктограмма помещена в меню ИНСТРУМЕНТЫ.

Оба эти инструмента так или иначе позволяют исправить оплошность, допущенную при работе, причем работа резиновой хорошо знакома ребятам в быту. В то же время ученикам достаточно трудно привыкнуть к команде ОТМЕНА, так как они никогда не встречались ни с чем подобным. Поэтому научить пользоваться командой ОТМЕНА — цель второй лабораторной работы.

Изучение инструмента КАРАНДАШ мы проводим в рамках творческой работы под названием «Грозные облака над штормовым морем».

Инструмент КАРАНДАШ имеет два дополнительных меню — ТОЛЩИНА и ЯРКОСТЬ и позволяет наносить произвольные линии различной толщины и насыщенности заданным цветом в соответствии с движением курсора по экрану. Указанные возможности инструмента помогают создавать достоверный пейзаж. Хорошо, если ребята перед выполнением творческой работы посмотрят работы Айвазовского, Куинджи и других художников.

Освоение инструментов КРАСКА, СИМВОЛЫ мы проводим в рамках изучения команды «Раскрась меня» из меню ДРУЖОК.

Инструмент КРАСКА предназначен для заполнения областей рисунка, ограниченных замкнутым контуром, сплошным цветом либо одним из 13 узоров, которые воспроизводятся на белом фоне. Варианты узоров представлены в дополнительном меню. При работе с инструментом имеются свои тонкости. Если вы хотите по красной крыше домика положить черную черепицу с помощью инструмента КРАСКА, то ничего не получится: черепица вытеснит красный цвет. А попытки закрасить другим цветом уже залитую узором поверхность не приводят к успеху, так как краска разольется в пределах одного фрагмента узора.

Однако возможность наложить узор на фон имеется, но при использовании другого инструмента, о чем ребята узнают позже.

В графическом редакторе KidPix существует набор файлов, каждый из которых представляет собой контурный рисунок какого-либо сюжета. Для того чтобы открыть такой файл, используется команда «Раскрась меня» из меню ДРУЖОК. Работая с этой командой, ребята

получают представление о хранении файлов в памяти ПК, об их считывании, а раскрашивая рисунок, они осваивают инструмент КРАСКА.

В конце работы мы предлагаем придумать название и нанести его на готовый рисунок. Для этого существует инструмент СИМВОЛЫ, который позволяет делать надписи печатными русскими или латинскими буквами и знаками одного размера. В дополнительном меню инструмента представлены два алфавита, цифры и знаки. При выборе того или иного символа его название произносится голосами с различной интонацией (по-детски, басом, хором, вкрадчиво, весело и т. д.). Такое представление режима вызывает неизменный интерес у ребят.

Можно, активизировав режим в альтернативном варианте, менять шрифт, но эта операция не русифицирована.

В лабораторной работе, посвященной изучению инструмента ЛИНИЯ, ученики выполняют ряд заданий, направленных на освоение дополнительных меню — ТОЛЩИНА и ЯРКОСТЬ и специальных приемов построения вертикальных, горизонтальных и линий под углом 45 градусов. Для приобретения навыков работы с инструментом ребятам предлагается при помощи курсора-прицела построить пучок лучей из одной точки; нарисовать снежинку; придумать алгоритм деления отрезка пополам.

Мы старались так построить задания при изучении инструментов, чтобы учащиеся поняли назначение дополнительного меню и особенности инструмента при работе мышкой, когда нажата клавиша SHIFT или ALT.

Дополнительное меню режима ПРЯМОУГОЛЬНИК предлагает варианты узоров заполнения прямоугольника, а нажатая клавиша SHIFT помогает нарисовать точный квадрат. Ребята строят коврик из вложенных прямоугольников с различными узорами заполнения и шахматную доску из квадратных клеток двух цветов. Предварительно мы обсуждаем алгоритм построения такой клеточной доски, чтобы построить ее с наименьшим количеством переключений цвета с одного на другой. Для этого надо пририсовывать клетки не ряд за рядом (при этом пришлось бы переключать цвет столько раз, сколько клеток на доске), а по диагоналям, что значительно сокращает время выполнения задания.

В графическом редакторе KidPix

принят оригинальный принцип прорисовки кругов и эллипсов, отличающийся от многих других редакторов. Обычно размеры круга определяются его радиусом, а именно удалением курсора от точки нажатия клавиши мышки, принимаемой за центр круга. Здесь же эллипс (круг) трактуется как фигура, вписанная в соответствующий прямоугольник (квадрат), диагональ которого определяется точками, где клавиша мышки была нажата и отпущена. Рисовать предварительно прямоугольник не обязательно, но всегда надо об этом помнить. Этот принцип оправдывает себя, так как с редактором могут работать дети, еще не владеющие понятиями центра и радиуса круга.

Лабораторная работа по освоению инструмента КРУГ содержит ряд заданий различной сложности. Закрепление навыков проходит в рамках творческих работ «Зимний пейзаж со снеговиком» и «Экзотическая бабочка».

Большую помощь при построении рисунков оказывает ребятам инструмент ШТАМП, изучению которого посвящен ряд лабораторных работ. Штамп — это рисунок какого-либо объекта. Инструмент ШТАМП позволяет выбрать из набора штампов, заложенного в редактор, любой объект и поместить его в свой рисунок. Дополнительное меню режима содержит 8 групп по 14 штампов в каждой. В редактор заложено два набора по 112 штампов в каждом: «My Stamps» и «My Stamps-COMPANION», которые можно сменить по мере необходимости один на другой. Редактор предоставляет возможность увеличивать штампы в 2, 4, 8 раз. Задания по освоению инструмента включают составление логических цепочек, ребусов, сборку из штампов, представляющих части одной фигуры, единого целого.

С помощью команды «Редактор штампов» можно вносить изменения в рисунок штампа. При выполнении этой команды на рабочем поле появляется окно, в котором выбранный штамп увеличен. Каждый пиксель выбранного штампа здесь представлен в виде квадратика. В рамках работы с этой командой ребята получают понятие пикселя как элементарной единицы графического изображения. Изучение команды «Редактор штампов» завершается творческими работами «Фирменные знаки» и «Язык жестов».

Последними в рамках изучения ос-

новных инструментов и команд редактора являются лабораторные работы по освоению режима ПЕРЕМЕЩЕНИЕ.

Режим ПЕРЕМЕЩЕНИЕ позволяет переносить части рисунка с одного места на другое. В дополнительном меню можно выбрать рамку, которая устанавливает размеры перемещаемого участка. Для получения навыков работы с этими рамками ребятам предлагается задание, которое напоминает сборку русской матрешки. В заранее заготовленном файле расположены прямоугольные фигуры по размерам рамок. Надо, подобрав рамку нужного размера, вложить эти фигуры одна в другую и составить из них единое целое.

Но наиболее важным в дополнительном меню является инструмент МАГНИТ. МАГНИТ позволяет перемещать участок произвольной формы; уменьшать размеры выделенного участка по длине или по ширине, т. е. не пропорционально, как допускает один из подрежимов режима СПЕЦЭФФЕКТЫ, а как бы «сузить» или «сплющить» выделенный участок; выделять фрагменты рисунка, активизируя при этом меню РЕДАКТОР.

С помощью команд меню РЕДАКТОР можно удалять, копировать и вставлять необходимое число деталей в данный рисунок и в другие рисунки, наносить фактуру на цвет, чего нельзя было делать только с помощью одного режима КРАСКА.

Возможность «сужения» и «сплющивания» ребята используют в творческой работе «Королевство кривых зеркал». Приемы копирования и вставки фрагментов, а также нанесение фактуры на цвет осваивают в лабораторной работе «Построение кирпичного дома с черепичной крышей». В ней ученики строят кирпичную стену дома, начиная с одного кирпича, методом последовательного укрупнения копируемого фрагмента, а затем наносят кирпичную фактуру на цвет. Таким же приемом пристраивается цветная крыша и наносится черепица. Алгоритм построения подробно оговаривается в теоретической части урока.

Инструменты КИСТЬ и СПЕЦЭФФЕКТЫ используются в комплексных творческих работах и специально не изучаются.

Получив первые навыки работы с инструментами редактора, учащиеся переходят к выполнению комплексных работ.

## Алгоритмический подход к работе с графическим редактором

При работе с графическим редактором мы ставим целью развитие у учащихся алгоритмического способа мышления. Тщательно обдумывая пути решения той или иной задачи, дети учатся разбивать ее на ряд более простых.

Создание алгоритма для решения задач какого-либо типа, его представление исполнителю в удобной для него форме — это творческий процесс. Сегодня можно наблюдать, как растет стремление продвигать алгоритмический способ в различных областях трудовой деятельности людей, как ширится класс задач, которые удается алгоритмизировать. Не являются исключением и задачи, которые мы предлагаем решить при помощи графического редактора KidPix.

Каждая лабораторная работа — это своего рода алгоритм. Ученик — исполнитель. Разнообразие предлагаемых задач позволяет познакомить учащихся со всеми видами алгоритмов: линейными, разветвляющимися, циклическими.

Иллюстрацией может служить серия лабораторных работ под условным названием «Задачи на построение».

Процесс обучения в школе, как и работа по готовому алгоритму, развивает у учащихся в основном исполнительские способности: действия согласно правилам, в соответствии с инструкциями, по раз и навсегда заведенному порядку. Когда мы предлагаем ребятам разделить отрезок на две равные части, они все как один называют единственный способ деления: с помощью двух окружностей. Когда мы просим сделать это средствами графического редактора KidPix, энтузиазм несколько утихает. После подсказки о дополнительных возможностях, которые дает использование клавиши SHIFT, учащиеся начинают поиск решения. Он, как правило, завершается успешно — отрезок делится вертикалью, опущенной из точки пересечения двух линий, проведенных из граничных точек отрезка под углами 45 градусов. На этом простом примере мы убеждаем своих учеников, что любое ограничение в средствах усложняет поставленную задачу (все как в жизни). Второе задание: по заданному горизонтальному отрезку построить вертикальный такой же длины. Споткнувшись на первом задании,

учащиеся уже знают, что речь идет не о стандартном решении! Затем мы предлагаем провести параллельные прямые под любым углом; определить центр окружности, что требует дополнительных построений; построить окружность с радиусом, равным заданному отрезку... Так, от простого к сложному, шаг за шагом, подключая к решению задач познания в геометрии, мы подходим к построениям, которые кажутся невозможными в рамках детского графического редактора!

Вот задачи, которые предлагаются в лабораторных работах:

- построить правильный восьмиугольник;
- построить равнобедренный треугольник;
- построить правильный шестиугольник с заданной стороной;
- построить прямоугольный треугольник по заданным гипотенузе и катету;
- построить прямоугольный треугольник по катету и прилежащему углу 60 градусов;
- построить треугольник по трем сторонам.

Перечисленные задания имеют комплексный характер и требуют хороших навыков в работе с редактором. Работы сопровождаются подробным алгоритмом построения и чертежом, подтверждающим правомерность некоторых решений. Наряду с этим предлагаются задания, в которых ученики самостоятельно разрабатывают алгоритм, проходя таким образом путь от исполнителя до творца алгоритма.

## Межпредметные связи и разнообразие задач, решаемых при помощи графического редактора

Огромное значение на своих уроках мы придаем развивающим играм. Они концентрируют внимание, развивают память, умение находить ошибки и просчеты и исправлять их. В совокупности эти качества и называются изобретательностью.

Можно выделить обширную группу работ под общим названием «Конструкторы». Их можно условно разделить на работы по созданию элементов конструктора (плоских и объемных деталей)

и работы непосредственно по конструированию.

Особо следует отметить, что заготовки для всех конструкторов делают ученики VII—IX классов в лабораторных работах. Целью таких работ является не просто подготовка деталей, но и обеспечение единого типоразмера этих деталей (их графической совместимости) с помощью специальных приемов размножения и геометрических построений. Дети пользуются знаниями, полученными на уроках геометрии и стереометрии, что обеспечивает непрерывность процесса обучения. Такая работа требует точности исполнения и особой ответственности, так как результаты работы будут «увечены» на жестком диске и использованы в учебном процессе. В этом воспитательное значение заданий такого типа!

В работах по конструированию требуется составить графическую композицию из готового набора деталей с помощью распространенного приема «выделить, скопировать, вставить».

Плоским конструктором могут пользоваться все учащиеся на этапе изучения режима ПЕРЕМЕЩЕНИЕ и работы с меню РЕДАКТОР. А малыши пользуются конструктором, реализуя алгоритмы, составленные на теоретических уроках. При этом формируются понятия, относящиеся к предмету «Информатика» — алгоритм, оптимальность алгоритма и т. п., а также развивается пространственное воображение и умение логически мыслить.

Объемное конструирование представлено работами:

- конструирование из кубиков по изображению общего вида;
- конструирование из кирпичиков по изображению общего вида;
- конструирование из кирпичиков по трем проекциям изображения;
- конструирование из традиционного набора типа «Юный строитель» (кубики, башенки, арки и т. д.).

Для конструирования из кубиков нужен только один куб, построить который не представляет труда после освоения инструментов ПРЯМОУГОЛЬНИК и ЛИНИЯ. Видимые грани этого базового элемента конструктора заливаются оттенками одного цвета, затем кубик копируется в «карман». Дальнейший алгоритм работы прост: с помощью только одной команды ВСТАВИТЬ из меню РЕДАКТОР сконструировать объект в точ-

ном соответствии с изображением на выданной карточке (рис. 1, 2).

Конечно, работа с объемным конструктором предваряется выяснением общих правил для таких построений. С ребятами V—VI классов мы на теоретическом уроке разбираем «по косточкам» несколько объемных построений с целью выяснения: сколько кубиков в нижнем ряду (с учетом тех, которые не видны), сколько на заднем плане и т. д. Кстати, с методом последовательной детализации (разбором «по косточкам»), как с основным приемом решения сложных задач, мы знакомим детей уже в этом возрасте.

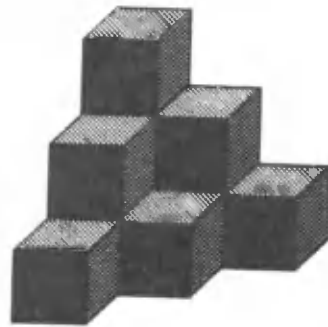


Рис. 1

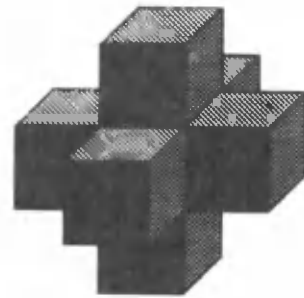


Рис. 2

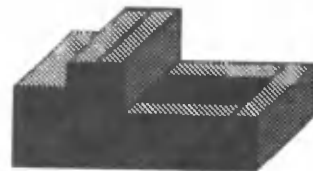


Рис. 3

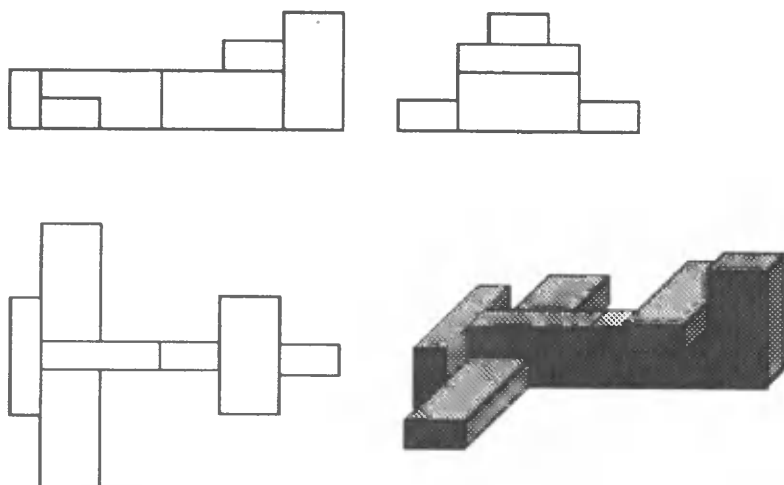


Рис. 4

Совместными усилиями мы выясняем основные правила: построения надо начинать с нижнего ряда, с дальнего плана и продвигаться слева направо!

Для конструирования из кирпичиков требуется изображение кирпичика в нескольких положениях (пашмя, на ребре, на торце) с согласованными размерами. KidPix не позволяет нам вращать фигуры, поэтому для хранения исходного кирпичика в разных положениях выделяется левая часть экрана. Мы назвали ее полем деталей, а свободную часть — полем конструирования. Готовый лист с полем деталей учащиеся вызывают из папки «Заготовки для лабораторных работ», чтобы сразу приступить к построению.

Объемные построения из кирпичиков по общему виду могут выполнять даже самые младшие ученики (рис. 3). Построения общего вида по заданным трем проекциям мы предлагаем в лабораторных работах для старшеклассников (рис. 4).

В качестве раздаточного материала на уроках по конструированию можно использовать рисунки из книги В. П. Никитина «Ступеньки творчества или развивающие игры» (серии К, КВ, СУ). Те, кто знаком с вышеназванной книгой, могут заметить, что игры рекомендованы детям дошкольного возраста! Но не забывайте, что играть с реальными кубиками и кирпичиками гораздо проще, чем с плоскостным изображением фигур на компьютере.

Ученики V—VII классов с удовольствием выполняют на компьютерах работы, которые мы называем «Игры со словами». Таких игр множество. Их полезность трудно переоценить! Они способствуют увеличению словарного запаса детей и их грамотности. Сами игры являются неотъемлемой частью занятий в теоретическом классе. Мы просто даем возможность использовать компьютер как подсобный инструмент для выполнения рутинной работы, связанной с вычерчиванием шаблонов для этих игр и заполнением их буквами. Ключевые слова и задания для таких работ либо выдаются учителем, либо являются домашней заготовкой. Описание этих игр можно встретить в любом детском журнале, и зачастую учащиеся сами находят и даже придумывают интересные словарные задачи.

Название «лабораторная работа» можно применить к таким играм с натяжкой, но, несмотря на сказочно-затейную оболочку этих заданий, они тоже выполняются по алгоритму и преследуют вполне определенные цели.

Кроме игр со словами в KidPix можно реализовать множество логических игр.

Серия заданий в графическом редакторе KidPix носит название «Одним росчерком пера». Специально подобранные рисунки, которые хранятся в папке «Заготовки для лабораторных работ», надо обвести инструментом КАРАНДАШ, не отпуская кнопку мыши и не проводя дважды по одной и той же линии. Задания сопровождаются леген-

дами и историческими анекдотами, связанными с определенными фигурами. Если с первого раза не получилось, надо командой ОТМЕНА удалить нарисованную линию и начать снова. Учитель при проверке пользуется этой же командой.

Выполнять задания «Одним росчерком пера» можно в самом начале знакомства с KidPix в рамках освоения инструмента КАРАНДАШ. Алгоритмы вычерчивания самих фигур (кстати, довольно сложные) реализуются гораздо позже в комплексных работах «Задачи на построение».

У самых маленьких освоение графического редактора проходит в игровой форме. Так, для освоения инструмента КРАСКА мы подготовили логические задачи «Раскрась симметрично» и «Найди одинаковые фигуры и закрась их одним цветом». С помощью режима ПЕРЕМЕЩЕНИЕ реализуются позиционные игры (передвижение фишек по игровому полю). Более того, редактор дает простор для творчества детей при оформлении таких игр. Список логических заданий безграничен, как детская фантазия: разнообразные логические цепочки, «черные ящики», ребусы, игры «Найди лишний предмет», «Найди различия в рисунке»...

### Творчество в работе с графическим редактором

Важное значение придаем мы творческой направленности занятий. Графический редактор KidPix помогает ребенку осознать в себе творца, дает возможность взглянуть на мир глазами создателя.

Когда в руки ребенка попадают карандаши, это уже пробуждает его творческую активность. Но если кроме карандашей под рукой оказывается набор первородных картинок, волшебная многоцветная кисть, лист копировальной бумаги, увеличительное стекло и другие удивительные инструменты, вокруг воцаряется атмосфера творческой приподнятости!

Наряду с лабораторными работами, которые предполагают четкое выполнение заданий по алгоритмам, в процессе изучения KidPix ученикам предлагается обширный перечень творческих работ. В них нет строгих алгоритмов, а лишь поставлена цель и оговариваются возможные средства ее достижения. Здесь есть

простор для детской фантазии, и в этом сходство подобных работ с сочинением на заданную тему. И в то же время ученики познают все новые возможности редактора и области его применения.

Тематика, сложность и трудоемкость этих работ должны учитывать возрастные особенности школьников. Надо не только подбирать задания так, чтобы на каждом занятии ребята узнавали что-то новое, но и стараться, чтобы они успевали осуществить работу в рамках выделенного времени и получить удовлетворение от ее завершения. Содержание их обговаривается на теоретическом уроке или непосредственно перед выполнением.

Например, работа «Грозовые облака над штормовым морем», проводимая в рамках изучения инструмента КАРАНДАШ. Казалось бы, какие трудности может таить этот режим? Но мы заметили, что дети практически не пользуются дополнительным меню ЯРКОСТЬ. И тогда родилась эта работа. Перед ней мы беседуем с учащимися о свойствах карандашей. В совместной беседе выясняется, что кроме цвета и толщины заточки у карандаша есть еще одно свойство — твердость. Почему линия, нарисованная мягким карандашом, такая яркая и сочная, а твердым — бледная, хотя цвет один и тот же? Мы даже рассматриваем линии, проведенные карандашами разной твердости из набора «Конструктор», под сильной лупой. Учитель поясняет, что от грифеля мягкого карандаша, при одинаковой силе нажатия на него, отслаивается больше чешуек (мельчайших частиц, точек), чем от твердого. После того как становится понятной суть меню ЯРКОСТЬ, дети могут приступить к работе над образом «Грозовые облака над штормовым морем» со знанием дела и с помощью этого меню и меню ПАЛИТРА создавать достоверные переходы полутонов и насыщенности цвета в пейзаже.

Работа «Составление рисунка ткани» предваряется беседой об этапах создания ткани. Создание ткани начинается с эскиза, выполняемого художником. Возможно, художник применяет на этой ступени графический редактор, подобный KidPix. А завершается работа изготовлением красящих валиков для каждого цвета, используемого в рисунке. Учащимся постарше можно рассказать, что некоторые технологические процессы, применяемые для создания красящих валиков (травление по фотшаблону),

сходны с процессом нанесения токопроводящих слоев в микросхемах. Таким образом у учащихся формируется убежденность в универсальности инструмента, которым они располагают.

В процессе беседы должен быть решен вопрос: какого размера эскиз делает художник? Красящий валик должен быть равен ширине окрашиваемой ткани. Но ведь узор на ткани, как правило, повторяется! В этом можно убедиться, рассматривая образцы ситца. Судя по идентичности повторяемых фрагментов, художник явно пользуется каким-то средством копирования, размножения. Цель конкретной работы и состоит в том, чтобы научиться многократно тиражировать небольшой фрагмент рисунка. В качестве средства достижения поставленной цели предлагается использовать дополнительное меню режима, условно названного нами СПЕЦЭФФЕКТЫ (пиктограмма «Миксер»). Этот режим не изучается на отдельном занятии, но на его уникальные возможности мы постоянно обращаем внимание школьников. Окно № 4 дополнительного меню дает возможность сделать сразу девять копий одного фрагмента, помещенного в центр экрана, одним щелчком мышки. Ребята сначала «репетируют» эту часть работы. Необходимо заметить, что если глазомер подвел ребенка и он «промахнулся», помещая рисунок в центр экрана, то его ждет неудача. Важно сразу подсказать учащимся, что, прежде чем щелкать, нужно, перемещаясь по экрану с нажатой кнопкой мыши, найти точку, в которой размножение удалось, и в ней отпустить кнопку. Следующий момент, который требует «репетиции», связан с использованием клавиши ALT вместе с вышеописанным приемом. Применение этой клавиши дает возможность стократного тиражирования!

В работе «Топографическая карта» можно использовать те же приемы размножения для получения таких объектов, как болото, лиственный и смешанный лес, луг. Но вот незадача! Тиражирование в режиме СПЕЦЭФФЕКТЫ требует целого экрана, ну а «карман» для копирования в меню РЕДАКТОР может хранить только одну копию! Где же хранить отдельные объекты топографической карты до компоновки? Ответ должен подсказать учитель — в реквизите «Альбом» (меню ЯБЛОКО). Сохранять рисунки в альбоме учащиеся к этому времени должны уметь. Новым является

то, что перед записью в альбом рисунок надо уменьшить в два раза. Делается это в том же режиме СПЕЦЭФФЕКТЫ. Уменьшенный рисунок появляется прямо на полноразмерном рисунке, и перед копированием его надо аккуратно обвести мигающей рамкой (режим ПЕРЕМЕЩЕНИЕ, подрежим МАГНИТ). После того как в альбом занесены все необходимые объекты, можно приступать к компоновке, и здесь пригодится и карандаш, и резинка, и умение передвигать фрагменты рисунка. Как и в работах по конструированию мы предлагаем ребятам сначала создать меню топографических объектов, перенес их из «Альбома» в левую часть экрана.

Важно следить, чтобы работы, связанные с общеобразовательными дисциплинами в школе, учитывали уровень подготовленности учеников и не опережали соответствующие уроки по содержанию. Работы «Топографическая карта» и «План местности» требуют домашних заготовок на листе бумаги для объективной оценки результата.

Работа «Перспективное изображение объектов на рисунке» проводится в рамках изучения инструмента ШТАМП. Перед выполнением работы необходимо поговорить о средствах передачи объема в плоских рисунках. Можно совершить краткий экскурс к наскальным рисункам древнего человека, которые не отображали объем, рассказать об экспериментах художников в этой области. Например, Анри Матисс, французский художник-импрессионист, динамику и объем в своих полотнах передавал сочетанием цветов (учащимся демонстрируются репродукции с полотен «Красная комната» и «Танец», находящихся в Эрмитаже). Но жизненный опыт подсказывает нам самый очевидный метод — уменьшение размеров объекта при его удалении от наблюдателя. Этот метод можно реализовать в работе, используя возможность двукратного (клавиша ALT), четырехкратного (клавиша SHIFT) и восьмикратного (клавиша ALT + SHIFT) увеличения оттиска штампа. Объекты для изображения в перспективе выбирают сами учащиеся. Как правило, это аллея, улица, пейзаж.

Творческая работа под условным названием «Королевство кривых зеркал» вызывает неизменный интерес у учащихся. Она проводится в рамках изучения подрежима МАГНИТ режима ПЕРЕМЕЩЕНИЕ. Кроме возможностей рабо-

ты с меню РЕДАКТОР, которые изучаются посредством специальных заданий в лабораторных работах, этот подрежим можно использовать для непропорционального уменьшения размеров рисунка. Если передвигать объект, обведенный волшебной магнитной рамкой, к границам экранного поля, то, натываясь на них, фрагмент начинает сплющиваться! Левая и правая границы сплющивают его по горизонтали, верхняя и нижняя — по вертикали. Инструмент ШТАМП содержит множество «разборных» изображений. Состыковывая разноувеличенные фрагменты одного рисунка, можно добиться разных эффектов искажения. При помощи этих нехитрых приемов и команд «Вырезать» и «Вставить» из меню РЕДАКТОР учащиеся создают галерею персонажей комнаты смеха.

Работы «Фирменные знаки» и «Язык жестов» проводятся для иллюстрации применения команды «Редактор штампов» меню СЕРВИС при изучении режима ШТАМП. Работы предполагают предварительное изучение окна «Редактор штампов». Перед учениками открывается еще одна возможность для реализации их творческих устремлений — возможность подправить имеющийся штамп или даже внести в библиотеку рисунков свой собственный. Как обычно, работам предшествуют беседы либо о фирменных знаках, эмблемах и их важности в сфере рекламы (примером может служить знаменитое надкусанное яблоко фирмы APPLE), либо о международном языке жестов, понятном каждому без всяких слов. Та или иная работа предлагается с учетом интересов конкретной группы учащихся и сопровождается, по их желанию, раздаточным материалом с изображением известных знаков и жестов.

Работа старшеклассников по созда-

нию штампов завершается сохранением самых удачных на страницах реквизита «Альбом» из меню ЯБЛОКО. Они используются в дальнейшем, при изучении текстовых редакторов в качестве иллюстративных вставок в тренировочные тексты «О языке жестов» и «Фирменные знаки». Таким образом, уже на стадии изучения KidPix ребята узнают о возможной транспортировке рисунков в другие программные среды.

Чтобы показать дополнительные, не свойственные другим графическим редакторам возможности KidPix, мы проводим работу «Озвучивание картинок» с использованием команды «Запись звука» из меню СЕРВИС. Эта работа поможет учащимся, освоившим режимы «Запись», «Пауза», «Воспроизведение», на уроках при работе с бытовым магнитофоном.

Работа «Композиция с мультипликационной вставкой» знакомит детей с возможностями команд «Веселый телевизор» и «Смотреть кино» из меню ДРУЖОК. Роль учителя здесь состоит в том, чтобы дать практические советы о последовательности действий при создании «живого» сюжета.

Список творческих работ можно было бы продолжить. Важно помнить, что эти работы должны соответствовать тематике проводимых на компьютерах лабораторных работ и уровню подготовленности учащихся на данный момент времени.

Хорошо организованная и тщательно продуманная деятельность ребенка на уроке помогает ему не только познать премудрости работы с графическими редакторами, но и научиться самостоятельно и последовательно доводить начатое дело до конца, проявлять инициативу и добиваться поставленной цели, невзирая на трудности!



# ПРОФЕССИОНАЛЬНЫЙ КОМПЬЮТЕР

М. А. Кузьменко,

*ИнфоМир*

## ПЕРСОНАЛЬНЫЕ КОМПЬЮТЕРЫ ВЧЕРА И СЕГОДНЯ

Считается, что первую электронную вычислительную машину (ЭВМ) или, как принято сейчас говорить — компьютер, сконструировали в 1945 г. в США. Значительную роль в создании ЭВМ сыграл американский ученый венгерского происхождения Джон (Янош) фон Нейман (1903—1957), который сформулировал принципы ее работы. Согласно этим принципам, компьютер располагается устройствами, которые выполняют хранящиеся в памяти программы и производят операции над данными. Устройство управления воспринимает команды программы и выполняет их, арифметико-логическое устройство (АЛУ) предназначено для вычислений. На современных компьютерах эти устройства включены в БИС, называемую центральным процессором.

Состояние и развитие ЭВМ всегда обуславливается фундаментальными достижениями электронной техники, особенно в области научных и конструкторско-технологических разработок элементной базы (электронных приборов и устройств), в значительной степени определяющей функциональные возможности, структуру и технико-эксплуатационные характеристики ЭВМ. Именно поэтому понятие «поколение ЭВМ» в первую очередь связано с поколениями элементной базы и физическими принципами ее реализации.

ЭВМ первого поколения (40—50-е гг.) строились на дискретных радио- и электрокомпонентах и электронно-вакуумных приборах — лампах. В запоминающих устройствах (ЗУ) применялись магнитные барабаны и электронно-лучевые запоминающие приборы. Надежность этих ЭВМ была невысокой. ЭВМ первого поколения были ориентированы преимущественно на численное решение научно-технических задач с от-

носительно небольшим количеством входной и выходной информации.

В ЭВМ второго поколения (50—60-е гг.) в качестве элементной базы использовались полупроводниковые приборы (транзисторы, диоды) и миниатюрные радиодетали (резисторы, конденсаторы, разъемы), а в качестве конструктивной основы — печатные платы на съемных ячейках. Носителями данных в оперативных ЗУ служили ферритовые сердечники. Внешние ЗУ выполнялись на магнитных лентах. Применение полупроводниковых приборов позволило существенно повысить надежность и значительно уменьшить потребляемую мощность и размеры ЭВМ. Помимо решения научно-технических задач ЭВМ второго поколения применялись для обработки плано-экономической информации и для решения управленческих задач.

Для ЭВМ третьего поколения (60—70-е гг.) кроме использования принципиально новой элементной базы (все логические элементы, оперативные и постоянные ЗУ выполнены на интегральных схемах) характерны модульный принцип построения, программная совместимость, наличие базового программного обеспечения, возможность выполнения нескольких программ одновременно, улучшение эксплуатационных характеристик и высокая надежность. В качестве носителей данных во внешних ЗУ использовались жесткие магнитные диски и магнитные ленты. К третьему поколению у нас в стране относились широко известные в свое время ЕС ЭВМ и СМ ЭВМ.

Следующее поколение ЭВМ — четвертое — связано с развитием микропроцессорной техники. В 1971 г. компанией Intel была выпущена микросхема i4004 — первый микропроцессор и ро-

доначалник абсолютно доминирующего и самого известного сегодня семейства. Вот история его создания. Японская компания Busicom, ныне уже не существующая, заказала Intel Corporation изготовить 12 типов микросхем для использования их в калькуляторах различных моделей. Малый объем каждой партии микросхем увеличивал стоимость их разработки. Однако разработчикам Intel удалось создать такое устройство — микропроцессор, который мог использоваться во всех калькуляторах. Его тактовая частота не превышала 0,75 МГц. Первый процессор был четырехразрядным, что позволяло кодировать все цифры и символы. Этого было достаточно, чтобы использовать его в микрокалькуляторах.

Однако серьезные машины работают не только с цифрами, но и с текстом. Для того чтобы закодировать все цифры, буквы и специальные символы, необходимо было увеличить разрядность процессора. В результате в 1972 г. появился 8-разрядный i8008, а в 1974 г. был разработан i8080. Этот 8-разрядный микропроцессор был выполнен по NMOS (N-channel Metal Oxide Semiconductor) технологии, а его тактовая частота не превышала 2 МГц. У него было более широкое множество микрокоманд. Кроме того, это был первый микропроцессор, который мог делить числа. Процессор i8080 оказал значительное влияние на дальнейшее развитие вычислительной техники.

Таким образом история развития электроники подошла к созданию персональных компьютеров. Во второй половине 70-х гг. сложилась благоприятная ситуация для их появления на рынке. Ощущалась потребность в недорогих ЭВМ, способных поддерживать одно рабочее место. Многие персональные компьютеры того времени базировались на 8-разрядных процессорах, таких, как i8080, и дальнейшей разработке компании Zilog Corporation — Z80. Стандартом операционной системы для персональных компьютеров стала разработанная компанией Digital Research CP/M (Control Program for Microcomputers). Она была сделана по образцу операционных систем больших ЭВМ, но размеры были гораздо меньше, что давало возможность работать на микропроцессоре.

Тем временем в 1976 г. была основана компания Apple Computer. История ее создания, ставшая легендой, многократно пересказана компьютерными изданиями, однако трудно удержаться от

хотя бы краткого изложения основных моментов. Итак, молодые инженеры Стив Возняк и Стив Джобс в гараже собрали первую партию сконструированных ими Apple, причем Джобсу пришлось продать свой автомобиль «Фольксваген». Apple-I не имел ни клавиатуры, ни корпуса. В апреле 1977 г. был выпущен Apple-II. В то же время появилась знаменитая эмблема в виде надкушенного разноцветного яблока. Apple-II имел одноплатную конструкцию и шину расширения, позволяющую подсоединять дополнительные устройства. Клавиатура была помещена в отдельный корпус, что соответствовало простому, практичному и эффективному подходу. В качестве центрального процессора был взят надежный по тому времени 8-разрядный 6502. Недостатком являлось отсутствие нижнего регистра символов, малая оперативная память — только 8 килобайт и способность обрабатывать лишь 40 столбцов символов на экране. Для увеличения объема памяти использовалась магнитная лента, запускаемая с обычного кассетного магнитофона. В дальнейшем к Apple-II были разработаны графические видеоадаптеры, дисковая операционная система для управления оперативной памятью и нижний регистр для символов, которые могли размещаться на экране в 80 столбцах. Со временем этот компьютер быстро завоевал рынок, постоянно совершенствовался, и за 10 с небольшим лет его выпуска было продано более 2 млн. экземпляров. Цена его колебалась около \$1000.

Своим коммерческим успехом Apple-II обязан в значительной степени открытости его архитектуры. Этот компьютер обладал модульной структурой с возможностью расширения системы. Техническая информация об Apple-II была опубликована, что дало возможность независимым производителям изготавливать совместимую аппаратуру.

К 1980 г. стал очевиден успех идеи персональных компьютеров. Их рынок достиг нескольких десятков тысяч в год. Крупнейшая электронная корпорация США International Business Machines (IBM), лидер в производстве компьютеров, уже совершила одну стратегическую ошибку, уступив рынок мини-ЭВМ компании Digital Equipment Corporation (DEC), и не хотела ее повторять на растущем и перспективном рынке ПК. Тем более причиной для беспокойства стал успех компьютеров компании Apple

Computer. IBM решает быстро захватить рынок.

Самым легким путем для завоевания рынка ПК была покупка. Финансовые возможности IBM позволяли просто купить завод по производству ПК. Однако IBM не хотела покупать чей-либо гараж, оптимистично названный заводом по производству компьютеров.

Сомнений не было: машина IBM должна быть реализована на микропроцессоре. Встал вопрос: какой микропроцессор использовать? Используемый в Apple процессор 6502 устарел уже в 1981 г. Он был 8-разрядным с частотой всего 1 МГц и, таким образом, проигрывал процессору Z80, который являлся также 8-разрядным, но имел более эффективный набор команд, и скорость его работы была выше. Однако для завоевания позиций IBM необходимо было создать существенно новый компьютер. И выбор был сделан в пользу процессора компании Intel — i8088.

В 1976 г. компания Intel начала усиленно работать над микропроцессором i8086. Размер его регистров был увеличен в 2 раза, что дало возможность увеличить производительность в 10 раз по сравнению с i8080. Кроме того, размер информационных шин был увеличен до 16 бит. Размер адресной шины в 20 бит позволял физически адресовать до 1 Мбайт оперативной памяти. В некотором смысле i8086 опередил свое время. ПК основывались на 8-разрядной архитектуре, память была дорога, требовались дополнительные 16-разрядные микросхемы. В 1979 г. был создан i8088. Он являлся близнецом i8086: 16-битные регистры, 20 адресных линий, тот же набор инструкций. Единственное различие состояло в том, что шина данных была уменьшена до 8 бит. Это позволяло использовать широко распространенные в то время 8-битные микросхемы. Первоначально процессор работал на частоте 4,77 МГц, но впоследствии другими компаниями были разработаны совместимые с ним процессоры, работающие на более высокой частоте — 8 или 10 МГц.

Итак, 12 августа 1981 г. IBM впервые представила свой первый персональный компьютер, который так и назывался IBM PC (Personal Computer). Он имел процессор i8088, два дисководов для гибких дисков по 160 Кбайт и оперативную память 64 Кбайт с возможностью расширения до 512 Кбайт. В ПЗУ PC был помещен язык программирования

Бейсик. IBM разработала свой собственный дисплей, который имел хорошую контрастность, символы на нем легко читались и не утомляли мерцанием. Были опубликованы программные и технические спецификации, которые позволили сторонним разработчикам создавать аппаратные расширения для IBM PC и программное обеспечение. В 1983 г. IBM выпустила новую модель PC XT (eXtended Technology) с жестким диском емкостью 10 Мбайт и оперативной памятью 640 Кбайт. Работали PC под управлением операционной системы MS DOS компании Microsoft, известной сегодня как крупнейший производитель программного обеспечения.

Невероятная популярность нового компьютера и значительные объемы продаж привели к тому, что уже в 1982 г. появились многочисленные их аналоги, созданные независимыми производителями. К 1984 г. IBM-совместимые компьютеры выпускали более 50 компаний, а в 1986 г. объем ежегодных продаж клонов превысил объем собственных разработок IBM, которая так больше никогда и не смогла вернуть себе лидерство. Архитектура IBM PC завоевала весь мир. Никакой другой оригинальной разработке, будь то Apple Macintosh, NeXT, Amiga или что-либо еще, не удалось занять место, сравнимое с IBM PC. Теперь название «IBM PC» принято использовать для всех поколений IBM PC-совместимых ЭВМ, а «XT» называют любой компьютер с процессором 8088 или совместимым.

Презентация IBM персонального компьютера AT (Advanced Technology) в 1984 г. сфокусировала все внимание на другом микропроцессоре — Intel 80286. Сам по себе микропроцессор был представлен еще в 1982 г. У него было много общего со своим предшественником i8086, но i80286 обладал такими дополнительными качествами, которые сразу привлекли пристальное внимание разработчиков, связанных с компьютерной индустрией. Он использовал 16-разрядную шину данных и 16-битные внутренние регистры. Первый i80286 работал на частоте 6 МГц, впоследствии были созданы модели, работающие на более высокой частоте (до 20 МГц). Первые AT были производительнее XT в 5 раз. В конечном счете самым большим преимуществом i80286 была способность работать с дополнительной памятью. Он имел 24-разрядную адресную шину, что по-

зволяло адресовать оперативную память 16 Мбайт. Intel 80286 мог также использовать виртуальную память до 1 Гбайт. Но в действительности на практике возможность преодолеть барьер адресуемой памяти в 1 Мбайт не была реализована. Для обеспечения совместимости с предыдущими моделями процессор мог работать в реальном режиме, который был похож на режим работы i8088/i8086, только быстрее. Кроме того, имелся защищенный режим, позволяющий адресовать память свыше 1 Мбайт. Процессор опередил свое время, и прошли годы, прежде чем появились OS/2 компании IBM и Windows компании Microsoft, работающие в этом режиме. Сегодня термином «АТ» называют все компьютеры, работающие на процессорах, совместимых с i80286. Компания Intel давно не выпускает этот процессор, как и более ранние модели, но старые их запасы еще иногда используются в специализированных контроллерах.

Тем временем в январе 1984 г. состоялась презентация первого компьютера Macintosh компании Apple Computer, появление которого сыграло значительную роль в развитии персональных компьютеров. Первые модели имели дизайн в стиле «all-in-one», т. е. системный блок и 9-дюймовый монитор в одном небольшом корпусе, массой около 20 фунтов. Компьютер занимал очень мало места на рабочем столе, число соединительных проводов было минимально. Небольшой размер экрана компенсировался чрезвычайно высокой четкостью изображения. В качестве центрального процессора использовался микропроцессор 68000 компании Motorola. Классические Macintosh — они так и назывались Classic — выпускались почти до настоящего времени. В поздних моделях использовался более мощный процессор Motorola 68030, а в некоторых был установлен дополнительно математический сопроцессор, а также цветной монитор. Такие компьютеры были удобны для домашнего применения, в учебных целях, для работы в офисе. К 90-м годам компания Apple Computer обошла IBM по объемам продаж персональных компьютеров. Но к тому времени IBM PC-совместимые машины выпускали очень многие...

В 1985 г. компанией Intel был анонсирован первый 32-разрядный микропроцессор i80386 (80386DX). Он имел все положительные качества своих пред-

шественников. Вся система команд i80286 полностью входила в множество команд 386-го. Новый процессор был полностью 32-разрядным и работал на частоте 16 МГц (впоследствии выпускались модели с частотой 25, 33 и 40 МГц). С увеличением шины данных до 32 бит число адресных линий было также увеличено до 32, что позволило микропроцессору обращаться прямо к 4 Гбайт (Гигабайт) физической памяти. Кроме того, он мог работать с 64 Тб (Терабайт=1024 Гбайт) виртуальной памяти. Огромное преимущество давал способ организации памяти. К ней можно было обращаться, как к одному большому полю, доступному для программ. Разделение памяти на сегменты по 64 Кбайт, как у i80286, было необязательно. Для того чтобы обеспечить совместимость с предыдущими микропроцессорами и программным обеспечением, i80386 был разработан таким образом, чтобы быть как можно более похожим на i8086 и i80286. Как и его предшественники, i80386 позволял работать в защищенном режиме с ограничением по адресуемой памяти в 1 Мбайт. В этом режиме он мог выполнять программы, разработанные на микропроцессорах предыдущих поколений. С реального режима (real mode) i80386 мог быть переведен в защищенный режим (protected mode). В противоположность i80286 новый процессор мог переходить из одного режима в другой без перезагрузки машины. Новый режим, названный виртуальным (virtual mode), позволял процессору работать как неограниченное количество i8086 в одно и то же время. Память разбивалась на несколько виртуальных машин, каждая из которых работала так, будто она была отдельным компьютером на i8086. Любая из этих виртуальных машин могла запускать свою собственную программу, которая была полностью изолирована от всех остальных. Таким образом, можно было одновременно выполнять несколько программ на одном компьютере. Однако полностью 32-разрядные многозадачные операционные системы и приложения появились намного позже, когда i80386 уже почти не использовался... (Поставки системы OS/2 Warp 3.0 начались в 1994 г., а система Windows-95 на момент написания обзора существует только в бета-версии.)

Первый компьютер на 386-м процессоре был изготовлен компанией Compaq Computers. В апреле 1987 г.

IBM, чувствуя, что под натиском производителей клонов теряет рынок, объявила о создании семейства персональных компьютеров PS/2 с шиной MCA (Micro Channel Architecture). До этого компьютеры PC AT использовали для связи между устройствами компьютера шину ISA (Industry Standard Architecture). Частота передачи этой шины — 8 МГц, разрядность — 16 бит. Новая шина MCA была 32-разрядной, работала на частоте 10 МГц и, кроме того, имела средства автоматического конфигурирования и арбитража запросов на обслуживание. Девять компаний-клонмэйкеров во главе с Compaq (AST, Epson, Hewlett-Packard, NEC, Olivetti, Tandy, Wyse и Zenith) ответили на это в 1989 г. разработкой шины EISA (Extended Industry Standard Architecture). Как и MCA, она имела разрядность 32 и обеспечивала автоматическое конфигурирование и арбитраж. В отличие от MCA шина EISA полностью совместима с ISA: она работает на частоте 8 МГц и допускает подключение любой платы ISA в любой разъем EISA. Ни одной из этих шин не удалось занять доминирующее положение на рынке. Компьютеры с шиной MCA не были совместимы с ISA, а периферийные устройства были мало распространены. Компьютеры с шиной EISA стоили довольно дорого.

В 1988 г. компанией Intel был разработан микропроцессор i80386SX. Стоил он дешевле, чем i80386DX, и использовал 16-разрядную внешнюю шину данных, что было особенно удобно для применения в стандартных PC AT с шиной ISA. Однако, в отличие от i80286, он имел 32-разрядные регистры и мог работать во всех режимах 386-го, что позволяло функционировать на нем программному обеспечению, написанному для i80386DX. Компьютер с работающим на частоте 20 МГц i80386SX был на треть производительнее самого быстрого 286-го.

Компания Intel выдала лицензии на производство процессора 80386 некоторым производителям, и со временем ее доля на рынке 386-х сократилась. А в 1989 г. появилась новая разработка Intel — микропроцессор i80486 (80486DX). Он объединял около 1200000 транзисторов (86 — 29000, 286 — 130000, 386 — 275000), что дает право говорить о наступлении пятого поколения компьютеров. Новый микропроцессор полностью совместимый по сис-

теме команд с предыдущими моделями семейства i80x86, кроме того, содержал в себе математический сопроцессор и 8 Кбайт кэш-памяти. Архитектура i80486 в сравнении с 386-й отличается многими серьезными улучшениями. Простые команды, например межрегистровые пересылки, выполняются им за один цикл, тогда как обработка этих команд i80386 занимает два-три. Интерфейс внешней шины i80486 более эффективен, чем интерфейс 386-го, который не поддерживает пакетную передачу данных и по этой причине во многих случаях работает медленнее. 386-й процессор, вообще не имеющий внутренней кэш-памяти, вынужден дожидаться поступления команд и данных, в то время как i80486 работает без задержек. Интегрированное скоростное устройство обработки чисел с плавающей точкой 486-го процессора по быстродействию намного превосходит аналогичное устройство 386-го. Блок сопряжения i80386 с дополнительным сопроцессором по сравнению с соответствующим блоком i80486 неэффективен и ограничивает производительность на операциях с плавающей точкой. Тактовая частота процессора — 33 МГц.

В 1991 г. компания Intel представила процессор i80486SX, отличавшийся отсутствием у него математического сопроцессора. А в 1992 г. — процессор i80486DX2, работающий с удвоенной внутренней частотой (66 МГц), но устанавливаемый на обычную материнскую плату. Впоследствии вышли процессоры с утроением частоты, названные почему-то i80486DX4.

Кроме компании Intel 486-е процессоры выпускают и некоторые другие компании, наиболее известные из которых Advanced Micro Devices (AMD) и Sunix. Чтобы избежать судебных преследований за нарушение авторских прав, они предусмотрели выпуск процессоров, отсутствующих в производственной программе Intel, например DX и SX на тактовой частоте 40 МГц, а также внесли в конструкцию некоторые усовершенствования. Тем не менее они считаются полностью совместимыми. В настоящее время цена на 486-е процессоры падает и составляет около \$150—160 за оригинальный Intel 80486DX2, работающий на частоте 66 МГц при напряжении 5 В.

Стандартом де-факто стало использование для 486-х систем локальной шины VL-Bus, предложенной ассоциацией

VESA (Video Electronics Standards Association). Применяя в компьютере локальную шину, позволяющую соединять процессор и периферийные устройства посредством широкого быстрого канала передачи данных, можно получить огромный выигрыш в быстродействии по сравнению с аналогичными компьютерами с шиной ISA. Шина VL-Bus является расширением шины 486-го процессора. Выводы процессора подключаются непосредственно к контактам разъемов шины. Тактовая частота шины равна тактовой частоте процессора либо, если в процессоре используется удвоение частоты, его половинной частоте. Пропускная способность шины — 132 Мбайт/с. При подключении каждой платы VL-Bus создается дополнительная нагрузка на выводы процессора, поэтому ограничиваются двумя-тремя контроллерами. Чаще всего это видеоакселератор и контроллер жесткого диска.

К 1992 г. во всем мире пользователи IBM-совместимой техники в основном перешли на компьютеры на базе 486-х процессоров. Именно с учетом его преимуществ создано многочисленное программное обеспечение.

Второе место по распространению после IBM-совместимых персональных компьютеров устойчиво занимает Macintosh компании Apple Computer. История их появления в России заслуживает отдельной статьи, здесь же перечислю основные семейства этих компьютеров. Я уже упоминал выше серию Classic. С 1987 по 1993 г. выпускались компьютеры серии II. Это были первые цветные Macintosh. Их архитектура стала открытой, что дало возможность их расширения. Выпускались модели II, IIx, IIcx, IIfx, IIfx, IIsi, IIsx, IIvi и IIvx. С 1990 г. выпускаются компьютеры серии LC — недорогие настольные компьютеры в плоском корпусе, получившие широкое распространение. Они подходят для использования в офисе, дома или в системе образования. Выпускались модели LC, LC II, LC III и другие. В настоящее время на базе процессора Motorola 68LC040 производятся LC 475, LC 575 и LC 630. Два последних могут комплектоваться дисководом CD-ROM.

До недавнего времени самыми высокопроизводительными Macintosh были компьютеры семейства Quadra. Они оснащены процессором 68040 с тактовой частотой до 33 МГц, сопроцессором, возможностью расширения ОЗУ до

256 Мбайт. Quadra используется в системах настольных типографий, в мультимедиа-приложениях и других задачах, требующих больших вычислительных мощностей и обработки значительных объемов данных. Наиболее подходят они и для создания программного обеспечения. Были выпущены модели Quadra 700, 900, 800, 950. Одно время наряду с Quadra выпускались компьютеры серии Centris. Также с 1993 г. выпускаются компьютеры подсемейства AV. AV-технологии объединяют на платформе Macintosh телекоммуникационные, видео и речевые возможности, обеспечивая пользователя этими основными способами взаимодействия и придавая им новые качества по сравнению с традиционными телефоном, телевизором и видеоманитофоном. Компьютеры Macintosh могут производить телефонные вызовы, записывать, отвечать на них. Имеется встроенная поддержка для организации телеконференций и даже видеоконференций. Macintosh серии AV имеют стандартные видеовыходы и видеовыходы, как видеоманитофон или телевизор, и автоматически настраиваются на международные видеостандарты. Пользователи имеют простой доступ к «живому» видео от внешнего источника в окне произвольного размера, могут выводить изображение как на телевизор, так и на монитор компьютера.

Кроме вышеперечисленных моделей Apple Computer выпускает портативные компьютеры серии PowerBook, завоевавшие любовь пользователей и отмеченные многими международными призами. Большое распространение получили компьютеры семейства Performa моделей 475, 575, 630, которые ориентированы на домашнее использование, оснащены факс-модемом и продаются с большим набором предустановленного программного обеспечения, необходимого для работы.

Очередной ступенью в развитии персональных компьютеров стало, как и ранее, появление новых, более мощных микропроцессоров, которые я рассмотрю подробнее. Это Pentium компании Intel и PowerPC — разработка консорциума Apple Computer, IBM и Motorola. Сначала о Pentium.

В 1993 г. компания Intel начала промышленные поставки нового процессора. Intel не стал присваивать новому процессору цифровое имя (например, 80586), которое нельзя было бы зарегистри-

стрировать как торговую марку, что исключало возможность использования без своего ведома. Поэтому новый процессор получил имя Pentium.

Первые модели работали на тактовой частоте 60 и 66 МГц и объединяли в себе до 3300000 транзисторов. Pentium полностью совместим со своими предшественниками. Тем не менее это первый 64-разрядный суперскалярный процессор с RISC-ядром, изготовленный по 0,8-микронной технологии BiCMOS. Основу процессора составляют два пятиступенчатых конвейера, аналогичных имеющемуся в 486-м, однако Pentium позволяет выполнять две команды за один такт синхронизации. Один конвейер позволяет выполнять любые операции, как целочисленные, так и с плавающей точкой, второй реализует часть целочисленных команд. Устройство для вычисления с плавающей точкой (FPU) представляет собой совершенно новую разработку, оно работает совместно с конвейерами команд, которые начинают обработку инструкций, а затем при необходимости передают их FPU. В целом же при выполнении команд используется сложная семиступенчатая конвейеризация. Основные операции — сложение, умножение и деление — реализованы аппаратно. Сочетание этих решений обеспечивает резкое повышение производительности Pentium. Ускорить вычисления за счет уменьшения частоты обращений к внешнему ОЗУ помогают два внутренних блока кэш-памяти — по 8 Кбайт для команд и данных. Двухканальная множественно-ассоциативная кэш-память данных обеспечивает одновременную работу с обоими конвейерами команд не только по чтению, но и по записи. Следующая новинка — система предсказания ветвления. При выполнении команды условного перехода в особой области памяти запоминается адрес перехода. При повторении этой команды процессор пытается «предсказать» результат ветвления. Если прогноз сбывается, переход происходит быстрее. Предсказание ветвления ощутимо повышает скорость выполнения циклов.

Впоследствии появились модели Pentium, работающие на тактовой частоте 90 и 100 МГц. Много шума наделало обнаружение ошибки в устройстве деления процессора. Компании Intel пришлось дать подробное описание этого дефекта и опубликовать результаты статистического анализа частоты возникнове-

ния сбоев для различных категорий приложений. Так, например, при использовании графических и издательских пакетов средняя расчетная продолжительность функционирования между отказами составляет 270 лет. Разумеется, ошибка со временем была исправлена, и сегодня в прайс-листах компаний, поставляющих комплектующие для компьютеров, можно встретить против позиции Pentium пометку BUG FREE!, т. е., буквально свободный от ошибки.

К маю 1995 г. на мировом рынке объем продаж компьютеров на базе процессора Pentium сравнялся с продажами техники с 486-ми процессорами, Pentium становится новым стандартом для IBM-совместимых компьютеров. Чаще всего в них применяют процессорно-независимую шину PCI (Peripheral Component Interconnect), разработанную компанией Intel. В отличие от шины VL-Bus, которая подключается непосредственно к процессору, PCI создает между процессором и периферийными устройствами некоторый промежуточный уровень. В стандарте PCI предусмотрены контроллер и акселератор, образующие локальную шину, не связанную с шиной процессора. Используются несколько способов повышения пропускной способности. Один из них — блочная передача последовательных данных. Если данные не являются последовательными, требуется дополнительное время на установку адреса каждого их элемента. Однако при передаче последовательных данных за счет блочной передачи адрес следующего элемента можно вычислять одновременно с чтением данных по текущему адресу.

Другой способ ускорения передачи данных — мультиплексирование — предусматривает передачу последовательных данных по адресным линиям, что позволяет удвоить пропускную способность канала. Наконец, акселератор PCI может накапливать информацию в буферах, что позволяет ему одновременно читать данные из памяти и вести блочный обмен с периферийным устройством. Шина PCI отделена от шины процессора и имеет усиленные сигналы, поэтому к ней может быть подключено больше устройств, чем к шине VL-Bus. Кроме того, PCI позволяет работать с контроллерами, использующими напряжение как 5 В, так и 3,3 В.

Другой новый микропроцессор, о котором я хочу рассказать, тоже по-свое-

му революционен. Речь идет о PowerPC, разработанном компаниями Apple Computer, IBM и Motorola. Промышленные поставки первой модели PowerPC 601 начались в конце 1993 г., а уже в 1994 г. компания Apple Computer стала использовать его в новых моделях компьютера Macintosh. Главным отличием PowerPC от используемых в IBM-совместимых компьютерах процессоров семейства i80x86 и используемых в Macintosh процессорах Motorola 680x0 является архитектура RISC (Reduced Instruction Set Computer), т. е. процессор использует сокращенный набор команд, в то время как предыдущие процессоры использовали сложный набор команд — CISC (Complex Instruction Set Computer). RISC производительнее CISC-процессоров за счет более быстрого выполнения команд. Все команды имеют одинаковую длину, что облегчает их выборку из памяти, и выполняются процессором за один такт. RISC-процессоры никогда не приостанавливаются для чтения дополнительных слов, образующих выполняемую команду. Сокращенный набор команд позволяет упростить схемы обработки. Простые способы адресации обеспечивают быстрый доступ к основной памяти. В RISC-процессорах не используются сложные вычисления.

В процессорах PowerPC не только реализованы перечисленные выше принципы RISC-архитектуры, но и предусмотрен ряд других средств повышения быстродействия. Например, PowerPC с помощью многокаскадного конвейера одновременно обрабатывает несколько команд. Блок распределения команд одновременно хранит восемь команд, три из которых каждый такт он может отправлять на одновременную обработку в разные исполнительные блоки. Кроме того, он имеет кэш-память 32 Кбайт, в которой хранятся часто используемые команды и данные, что уменьшает задержки, связанные с чтением из основного ОЗУ. Шина данных 601-го процессора 64-разрядная.

Следующие модели PowerPC — 603, 604 и 620 еще более производительны и имеют пониженную потребляемую мощность. В целях энергосбережения любой незагруженный исполнительный блок отключается и затем автоматически включается, как только потребуются вновь.

Компания Apple Computer взяла курс на производство компьютеров на

базе только PowerPC, эти модели стали называться PowerMacintosh. Они полностью совместимы с предыдущими моделями, «старые» программы выполняются в режиме эмуляции процессора 680x0. Эмулятор, «зашитый» в ПЗУ компьютера, интерпретирует систему команд процессора 68LC040. Также PowerMacintosh могут выполнять Windows-программы с помощью пакета SoftWindows компании Insignia Solution. (Истинные противники Macintosh в этом месте иронично заметят, что эмулировать Mac-программы на других платформах почему-то никому в голову не приходит.) Однако, для того чтобы ощутить все преимущества нового процессора, необходима разработанная с учетом его особенностей операционная система (System 7.5 отвечает этому требованию) и прикладные программы. Скорость выполнения программ, написанных в кодах PowerPC, в несколько раз превышает скорость работы программ на обычных моделях Macintosh. В настоящее время продаются PowerMacintosh на базе PowerPC 601 6100/66, 7100/80, 8100/100. Цифра в знаменателе обозначает тактовую частоту процессора. В 1995 г. был представлен Power Macintosh, работающий на частоте 110 МГц. Новые модели будут оснащены процессорами PowerPC 604 и 620 и шиной PCI.

Компания Apple Computer перевела на процессоры PowerPC и компьютеры семейства Performa. Их оснащение, включающее дисковод компакт-дисков, цветной 15-дюймовый монитор, 8 Мбайт ОЗУ, жесткий диск емкостью по меньшей мере 250 Мбайт и факс-модем, на сегодняшний день можно назвать достаточным.

Исход борьбы за господство на рынке персональных компьютеров пока не ясен. Куда пойдет компьютерный мир — покажет будущее. Пока оценки специалистов полярны. Одни считают, что Intel сохранит лидерство на рынке процессоров, а число пользователей IBM-совместимых компьютеров, благодаря широкому распространению программных продуктов компании Microsoft, приумножится. Другие полагают, что произойдет коренной поворот к компьютерам на базе процессоров PowerPC. Многое будет зависеть от того, какие новые решения предложат создатели аппаратуры, как будет развиваться программное обеспечение, а также от действий ведущих компаний на рынке.



Что же предпринимают лидеры компьютерной индустрии сегодня?

Компания Intel с целью упрочения своих позиций на рынке стала выпускать материнские платы для процессора Pentium. Также идет работа над выпуском новых моделей этого процессора, работающих на частотах 120, 133 и 150 МГц. Они изготавливаются по 0,4-микронной технологии. Ведутся разработки процессора нового поколения, условное название его — P6. Он будет работать на тактовой частоте 133 МГц при напряжении 2,9 В. В нем заложены такие функции обеспечения целостности данных, как обнаружение и коррекция ошибок, а также анализ и восстановление после неисправности. P6 поддерживает систему команд семейства 80x86, имеет встроенный кэш 256 Кбайт. Увеличению производительности способствуют исполнение команд с опережением и предсказание ветвлений. Микропроцессор P6 объединяет в себе 5500000 транзисторов. Intel совместно с Hewlett-Packard работают над созданием 64-разрядного процессора P7. Основная цель создания нового процессора — передача центральному процессору многих функций, пока осуществляемых дополнительными устройствами. Также совместно с Hewlett-Packard Intel разрабатывает RISC-процессор будущего PA-9000.

Компании Sugh и AMD разрабатывают процессоры класса Pentium — M1 и K5, соответственно. В связи с тем что Intel постепенно отказывается от развития линии 486-х, AMD разрабатывает более производительные модели процессора 486DX4, работающие на тактовой частоте 120 и 133 МГц. AMD достигла соглашения с Intel о прекращении взаимных правовых споров, а получила постоянную лицензию на микрокоды процессоров 386 и 486 и согласилась с тем, что не имеет права копировать микрокоды процессоров Pentium и P6.

Компания Apple Computer в течение многих лет отказывалась предоставлять лицензии на изготовление Macintosh-совместимых компьютеров, но сейчас пересмотрела свои позиции, и летом 1995 г. появился первый клон Macintosh. Первым обладателем лицен-

зии стала американская компания Radius Inc., крупнейший разработчик мониторов и другого периферийного оборудования для Macintosh. Первая модель Macintosh-совместимого компьютера от Radius называется Radius System 100.

Компания IBM работает над компьютерами на базе PowerPC и продолжает разработки операционной системы OS/2 для него под названием Warp for Power.

Компания Microsoft неоднократно откладывала сроки поставки окончательной версии операционной системы Windows-95, ранее условно называвшейся Windows 4.0 или Chicago. По последним сведениям поставки должны начаться 24 августа 1995 г.

За пределами этого обзора остались все еще используемые в наших школах БК, ДВК, УКНЦ, собранные на отечественном аналоге процессора компании Digital Equipment Corporation, применяемого в мини-ЭВМ типа PDP-11 (СМ ЭВМ), «Корветы» с операционной системой типа CP/M, на базе аналога Intel 8080, «Агаты» и прочая техника. Эти представители программно и аппаратно несовместимы между собой, а все вместе — с повсеместно распространенными в дивилизованном мире Macintosh и IBM PC. Кроме того, почти все они оснащены некачественными мониторами, а зачастую и вообще бытовыми телевизорами.

Пока же пользователям этих «компьютеров» как-то удается выживать, во многом благодаря энтузиазму некоторых отечественных компаний, не желающих терять обширный рынок и предлагающих или сделать IBM из «Корвета» или БК (Citroën из «Жигулей» или «Запорожца») или «новейшее» программное обеспечение. Не поднимается рука бросить в них камень за это. Но переход к современной технике необходим и неизбежен, как смена времен года.

*При подготовке обзора были использованы материалы КомпьютерУорлд-Москва/КомпьютерУик-Москва, Мир ПК, Компьютер Пресс, Коммерсантъ Daily.*

**С. И. Шумов,**

*старший научный сотрудник Института автоматизации проектирования РАН*

## ТЕЛЕВИЗОР МОЕЙ МЕЧТЫ

### Вступление

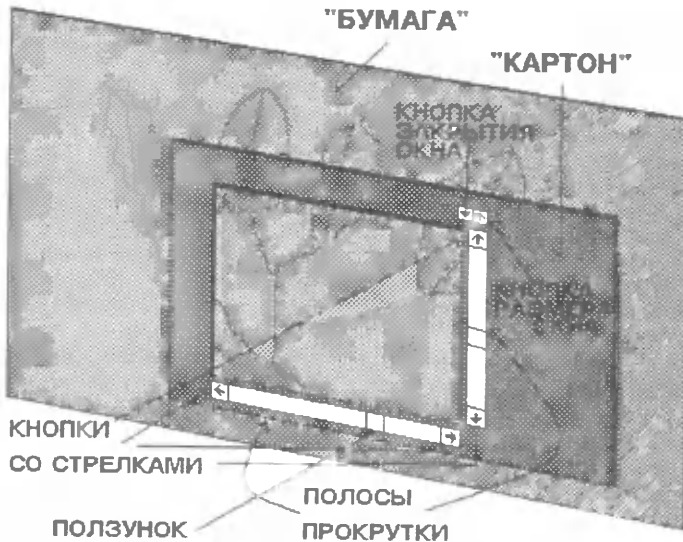
Когда я включаю телевизор, то так и хочется видеть на экране сразу все программы: шесть, семь, восемь... все, которые сейчас передают. И чтобы сразу можно было подключить звук той программы, которая интересна в данный момент, а если надо, то и развернуть окошко с ее изображением на весь экран... Словом, хорошо бы иметь Windows на телевизоре!

Увы, таких телевизоров еще нет, но уже продаются те, которые позволяют видеть на экране кроме основной еще одну программу в небольшом кадре.

А вот компьютеры уже похожи на телевизор моей мечты. И это благодаря Windows. Экран разделяется на несколько частей, и в каждом образовавшемся «окошке» выполняется своя программа\*. Правда, и здесь еще не дошли до совершенства: пока одна программа идет в активном окне, другие, как правило, ждут своей очереди. Но и здесь тенденция такова, что все они должны работать

одновременно, показывая результаты выполнения в отдельном окне. Отсюда и название системы Windows. Такое имя дала ей компания Microsoft, разработавшая эту систему и проделавшая огромную работу по ее внедрению.

Рассмотрим подробнее экран монитора компьютера, работающего под Windows. Каждое окно на экране имеет рамку, вдоль нижней и правой границы которой могут располагаться полосы прокрутки. Они позволяют перемещать изображение в окне вверх-вниз, влево-вправо. Для того чтобы понять, что именно при этом мы видим, представим, что экран монитора сделан из картона, в нем проделано прямоугольное отверстие-окно, а за окном располагается бумага с изображением или текстом. Поскольку бумага прижата сзади к картонному экрану, то видно только то, что не закрывает прямоугольное отверстие. Если теперь картон подвигать туда-сюда, то появится новый фрагмент того, что изображено или написано на бумаге. Для перемещения «картона» и существуют полосы про-



\* Появляются также и компьютеры, которые позволяют принимать телевизионные изображения. Пока, увы, можно смотреть одновременно только один телеканал. Но это только начало.

крутки. Каждая полоса содержит две кнопки со стрелками, а также бегунок, который позволяет знать, какую часть бумаги мы видим. При нажатии на какую-либо кнопку со стрелкой картон перемещается в направлении, указанном стрелкой: влево, вправо, вверх или вниз. Одновременно в этом же направлении будет перемещаться и бегунок. Как только он дойдет до нажимаемой кнопки, перемещение окна закончится: мы дошли в этом направлении до края бумаги.

Может статься, что ширина и высота бумаги совпадут с шириной и высотой окна. В этом случае перемещать окно не нужно, и вдоль границ окна Windows не показывает ни горизонтальную, ни вертикальную полосу прокрутки.

Теперь представим следующую ситуацию: у нас имеется набор географических планов или карт одной и той же местности, но выполненных в разных масштабах. Разумеется, размеры этих карт будут различны, поскольку мы предполагаем, что изображена одна и та же местность, т. е. географические координаты (широта и долгота) углов карт совпадают. Рассматривая через отверстие в картоне эти карты, мы будем видеть изображения местности с разной степенью подробности. *Большинство программ, работающих в операционной среде Windows, позволяют рассматривать изображения в разных масштабах.*

Размеры отверстия в картоне можно уменьшать вплоть до нуля и увеличивать до размера экрана. Размер изменяется перемещением правой или нижней стороны отверстия-окна. Есть еще кнопка мгновенного расширения окна до размера экрана, которая в распахнутом окне превращается в кнопку мгновенного уменьшения окна до прежнего размера.

Но это еще не всё: можно проделать несколько отверстий, и в каждом рассматривать свою бумагу. При этом одна бумага не будет мешать перемещаться и быть видимой другой. Да и отверстия могут частично или полностью совпадать. Здесь уже наступает черед выяснять, какое из них главнее, т. е. находится «ближе» к пользователю, и, соответственно, показывать бумагу в первую очередь в ней. Еще более «навороченные» компьютеры позволяют не только создавать окна на экране, но и перемещать сам экран, в нашей аналогии это значит двигать картон относительно монитора.

В общем, аналогия с бумагой и картоном с отверстием едва ли теперь стано-

вится применимой для объяснения других особенностей Windows. Пора переходить к другим сравнениям.

## Иконы, но не в церкви

Помимо собственно окон для Windows характерны пиктограммы, их еще называют значками или иконами. Последнее слово — прочитанное по буквам английское слово *icon*, что в переводе и означает: пиктограмма или значок. Значок — это маленькая картинка с подписью или без, которая наилучшим образом характеризует смысл того, что она означает. А означать она может программу или нечто созданное какой-либо программой: документ, рисунок, базу данных и т. д.

Значки располагаются в окнах. Если подвести курсор мыши к значку и дважды щелкнуть по нему левой клавишей, то произойдет инициализация какой-либо программы, скажем текстового редактора. Если активизируется иконка, связанная с созданным документом, то вызывается, например, тот же текстовый редактор, но уже с документом в окне, готовым для редактирования. В окнах есть еще кнопка, которая позволяет прекратить работу с приложением и превратить его опять-таки в значок.

## Windows — это вам не командир Нортона (или Волков)

Бытует мнение, что Windows представляется чем-то вроде чересчур красивого Нортона-командера. Автор одной достаточно интересной книги по программированию даже писал, что, дескать, часики в виде циферблата со стрелочками на экране — это хорошо, но по мне уж лучше иметь цифирки на мониторе, так даже современнее, а программы под Windows ну уж очень медленно работают, да и вызывать их можно, только написав имя программы и нажав после этого клавишу Enter\*. Лет шесть назад это было еще справедливо. Вспомним, что это было за время.

Компьютер на базе процессора Intel 80386 был пределом желаний, доступным немногим. Официально КОКОМ запрещал экспортировать в СССР компьютеры на процессорах с высокой, по тогдашним меркам, производитель-

\* Эта клавиша первоначально называлась CR — Carriage Return, возврат каретки. Название пошло от аналогичной клавиши электрической пишущей машинки.

стью. Некоторые солидные государственные (а тогда других не было) организации закупили наиболее современные компьютеры через третьи страны. Но даже и на них пакет Windows работал медленно. Только с появлением и широким распространением 486-го процессора Windows начал победоносно шествовать по планете.

Что значит медленно? Это можно передать через субъективное ощущение раздражающе медленно, когда приходится по несколько секунд ждать, пока обновится (т. е. «перерисует») экран вследствие какой-либо команды, ну, например, прокрутки текста или изображения в «окне». Верно и обратное: если происходит «почти мгновенное» обновление экрана в ответ на какую-либо команду, то компьютер работает «быстро». Более сложные программы требуют более быстрых компьютеров.

Скорость работы компьютера определяется различными факторами. Основной показатель — тактовая частота процессора. Например, процессор с частотой 100 МГц выполняет, как сказали бы раньше, 100 миллионов операций в секунду, т. е. имеет возможность за секунду 100000000 раз совершить какие-либо изменения в ячейках памяти компьютера, куда-то заслать ноль, куда-то единицу... Ясно, что чем выше частота, тем быстрее вроде бы должен работать компьютер. Но ведь работает в компьютере не только центральный процессор, но и множество других устройств, например накопители на дисках, устройства передачи и обмена информации и так далее. Они должны быть «под стать» центральному процессору, должны успевать за его работой и не давать ему простаивать.

Можно вспомнить такой пример. В середине 80-х годов в Москву приехал руководитель фирмы Apple и представлял новый компьютер Macintosh с интерфейсом типа Windows. Его процессор был по производительности примерно равен 386-му фирмы Intel, но оконный интерфейс работал куда быстрее. Дело в том, что программное обеспечение интерфейса на компьютерах Macintosh «зашиито» в постоянное запоминающее устройство, которое работает быстрее, чем память на диске.

Кстати говоря, оперативная память, RAM (Random Access Memory), также работает значительно быстрее памяти на магнитных дисках. Windows постоянно обращается к памяти, и если не хватает RAM, то используется жесткий диск. При работе Windows постоянно записывает на диск или считывает с него информацию, и это происходит тем

чаще, чем меньше RAM доступно системе. Дисковый накопитель намного медленнее оперативной памяти, отсюда следует вывод: чем меньше размер RAM на компьютере, тем медленнее работает на нем Windows. На практике на компьютере для работы Windows версий 3.1 или 3.11 требуется не менее 8 мегабайт оперативной памяти.

Кроме того, существуют жесткие диски как с быстрой, так и с медленной скоростью записи-считывания. Помимо этого на производительность Windows влияют такие факторы, как кэш-память, видеопамять, скорость обмена с системной шиной, а также и многое другое. Так что скорость «молотилки», т. е. тактовой частоты процессора хотя и влияет на производительность, но пропорционально, поэтому удвоение тактовой скорости не приведет к удвоению скорости работы любой компьютерной программы, в том числе и Windows.

Переход к новому поколению процессора дает выигрыш в производительности еще и за счет того, что работа процессора основывается на новых, более быстрых алгоритмах выполнения простейших операций: сложения, умножения, сдвига и т. п. Увеличивается разрядность чисел, что дает возможность производить вычисления с большей точностью. (Скорость света составляет 300000 км/с. За один компьютерный такт свет проходит  $300000/100000000 = 0,003$  км = 3 м.)

## Что дает Windows пользователю

Как известно пользователь — это тот, кто пользуется прикладными программами компьютера. Это могут быть как «серьезные» приложения (текстовые редакторы, электронные таблицы), так и игрушки.

Еще совсем недавно каждый программист делал интерфейс по своему вкусу. Меню располагались и сверху, и снизу, и сбоку экрана. Вход в меню в одних программах осуществлялся нажатием клавиши косой черты (Lotus 1-2-3), в других — F9 (Norton Commander), F10 (Лексикон), F3 (Word Perfect), Ins (Framework), Alt (Works), Esc (ChiWriter) и т. д. Кто-то сумел использовать мышь, кто-то нет. Кстати, такая ситуация до сих пор наблюдается в отечественных бухгалтерских программах.

В результате всего этого разноречия каждую новую программу нужно осваивать заново. Скажем, привык нажимать кнопку помощи Alt-h, а тут на тебе — F1 или F3, а то и вообще нет никакой подсказки. Привычка нажимать

определенные кнопки для выполнения определенных функций часто тормозит внедрение новых, более прогрессивных программных продуктов.

С течением времени интерфейс более или менее стандартизовался де-факто и де-юре. Не последнюю роль здесь сыграла система Windows. Познакомившись с ней, программисты стали писать свои интерфейсы в этом же стиле даже в DOS-программах. Кроме того, пользователю Windows не обязательно уметь читать, чтобы в ней работать, поскольку надо уметь только выбирать пиктограммы. Этот тезис подтверждает мой шестилетний сын, успешно освоивший основы работы в «окнах».

Стандартизация привела к тому, что разработчики программ начали искать новые метафоры для своих продуктов. Однако чаще всего осуществляется имитация традиционной технологии на экране, например показ красивого блокнота с вырезами для букв алфавита, или магнитофона с клавишами, или тех же часов с циферблатом. Кому-то успели надоеть песочные часы — символ ожидания.

Помимо всего прочего операционная система Windows решила проблему совместимости с периферийными устройствами, в первую очередь с принтерами. Так как если ведущие производители программного обеспечения еще старались, чтобы их программы были совместимы с большинством моделей принтеров, прилагали драйверы\* для работы с ними, то самостоятельные программисты и мелкие фирмы в своих программах нередко предлагают ограниченный выбор: широкий принтер или узкий, девяти- или двадцатичетырехугольный. Поэтому часто принтеры с такими программами не работают.

В Windows все эти проблемы решены раз и навсегда. Во время установки Windows на диск необходимо указать какой именно принтер подключен к вашему компьютеру. После этого любая программа под Windows будет печатать правильно на этом принтере. Программистам не надо заботиться о совместимости своих программ со всеми возможными типами печатающих устройств. Пусть производители «друкарков»\*\* ду-

мают о том, как совместить свой принтер с Windows. Разумеется, все Windows-программы уже не содержат все возможные драйверы для различных типов принтеров.

Попутно решилась и проблема русификации (украинизации, якутизации...) экрана и принтера. Национальные алфавиты легко устанавливаются в Windows, для многих языков существуют свои версии этой операционной оболочки. Для экрана и печати используются одни и те же шрифты, поэтому что видим на экране, то будет и на бумаге, поскольку и экран и принтер работают в графическом режиме.

### Что дает Windows программисту

Любой программист желает написать «фирменную» программу: различные меню, рамочки, хелпы, диалоговые окна. Тот, кто пытался это сделать сам на языке программирования, работающего под управлением операционной системы DOS, часто так и не мог закончить работу. Так как выяснялось, что дело это очень утомительное, требует много времени, а собственно программа, ее уникальная часть, составляет небольшой процент от трудозатрат при создании всего проекта. Кое-кому удавалось-таки написать собственный программный интерфейс, потом эти программисты использовали наработанные подпрограммы для других своих приложений. Если получались очень хорошие подпрограммы, то автор задумывался над их коммерческим использованием. Так появились Zups и ряд других. Некоторые ведущие производители программного обеспечения стали включать подобные оболочки в комплект языков программирования (Turbo Vision).

Разработчики Windows пошли еще дальше. Они включили интерфейс пользователя прямо в оболочку. Теперь не нужно иметь в своей программе процедуры рисования окошек и меню, все это уже есть в Windows. Достаточно указать наименования пунктов меню, расположение диалоговых элементов и их спи-

\* Драйверы — это такие программы, которые самостоятельно, как правило, не работают, а только дают возможность другим программам правильно работать с различными периферийными устройствами: принтерами, мониторами, клавиатурами, манипуляторами.

\*\*Этот неологизм взят из статьи в газете «Поиск», в которой говорилось о том, что в Беларуси пытались найти белорусские эквиваленты в научно-технической лексике, в частности предлагалось называть принтер друкаркой.

ски выбора, как Windows выдаст все это на экран в лучшем виде.

### Казалось бы, забытый интерпретатор

Многие еще не забыли, что на языке Бейсик строки надо нумеровать, и работает этот язык как интерпретатор, т. е. выполняет программу строчку за строчкой, каждый раз как бы заново ее читая, пока она не завершится успешно или не встретится ошибка в операторе. Программы в режиме интерпретации работают медленнее, чем оттранслируемые модули. По этой причине сейчас в основном используют трансляторы, в том числе и для Бейсика.

Новое поколение процессоров позволяет даже интерпретаторам работать быстро, поэтому интерпретаторы получили шанс открыть второе дыхание. В Windows интерпретация осуществляется по-своему, а именно подпрограммы интерфейса, работы с принтерами и другими периферийными устройствами, содержащиеся в Windows, находятся в динамически загружаемых библиотеках (dynamically linked libraries). Имена файлов с этими библиотеками имеют расширение dll. При работе Windows-программа по мере необходимости обращается к нужной библиотеке, как к той, которая находится в комплекте самой системы, так и к уникальной, написанной программистами специально для этой программы. Тем самым появляется возможность использовать одни и те же подпрограммы, не вставляя их собственно в программный код, что значительно уменьшает длину программы. Сами подпрограммы могут использоваться различными программами.

### Windows и объектно-ориентированное программирование

Для того чтобы использовать готовые подпрограммы и описания, появился особый стиль программирования, названный объектно-ориентированным (ООП). В этом случае описания переменных (объекты) и подпрограммы (методы) составляют единое целое, их можно легко модифицировать, создавать себе подо-

бные, копировать, заменять какие-то функции. Описание ООП выходит за рамки данной статьи. Отметим лишь одну особенность этого стиля: возникли специальные объектно-ориентированные языки программирования, использование которых достаточно сложно, так как имеются очень утяжеленные конструкции, поэтому переходят к автоматизации написания кода программ (языки четвертого поколения, CASE-технология).

### Окна не только в Windows

Несколько лет назад компьютерная общественность наблюдала за ходом судебного процесса, чтобы понять, что было раньше: интерфейс Windows или Macintosh. Пришли к выводу, что первоначально это сделала третья фирма. Есть Solaris — графическая оболочка в операционной системе UNIX, есть OS/2 фирмы IBM и т. п. Разрабатывается отечественный графический интерфейс.

Вернемся к тому, с чего мы начали статью, — к телевидению. Во времена моего детства видеотелефон считался светлым будущим всего человечества. Жители Москвы, Ленинграда и Киева могли общаться между собой при помощи этого средства связи, если им не лень было пойти на центральный переговорный пункт. Сейчас видеотелефоны стали появляться чаще, но никакого шума вокруг этого не происходит. Наверное, потому, что зачастую изображение не только не нужно, но и вредно. Не всякая женщина, например, рискнет подойти к видеотелефону, если она плохо выглядит. К обычному телефону — пожалуйста...

Но мы отвлеклись. В настоящее время видеотелефонную связь чаще всего называют телеконференцией. Собеседников может быть много — сколько уместится на экране окошек с лицами. Разумеется, эти физические лица могут находиться где угодно, хоть на разных краях света.

Итак, телевизор под Windows, телефон под Windows, на очереди стиральная машина. И это не шутка. Основатель фирмы Microsoft Билл Гейтс ставит целью, чтобы Windows были универсальным интерфейсом различных приборов, в том числе и бытовых. Чтобы не настоящие кнопки нажимать и ручки вертеть, а виртуальные, на экране.

# ТОЧКА ЗРЕНИЯ

А. Н. Гриценко, А. Н. Тушев,

г. Барнаул

## ВОЗМОЖНОСТИ И ПЕРСПЕКТИВЫ КЛАССОВ УКНЦ В УЧЕБНОМ ПРОЦЕССЕ

Терминальные классы УКНЦ («Электроника 0511») продолжают оставаться основным видом учебного оборудования кабинетов информатики средних учебных заведений. Говорить о их скорой замене на классы типа IBM пока не приходится. В связи с этим естественно возникает вопрос: «Возможно ли на УКНЦ вести учебный процесс на уровне современных требований и если да, то как этого добиться?» Данная статья представляет собой попытку дать ответ на этот вопрос. Высказанные ниже выводы и рекомендации — результат многолетней работы авторов статьи с классом УКНЦ, а также богатого опыта общения с учителями и разработчиками программного обеспечения. Авторы не претендуют на непререкаемость своих суждений и выводов, но все же надеются, что эта статья будет полезна пользователям классов УКНЦ.

### 1. Чего можно добиться с помощью класса УКНЦ

Основная задача курса информатики в школе — подготовка учащихся к будущей жизни и работе в информационном обществе. Следует ли отсюда, что для решения этой задачи необходимо немедленно установить в школах компьютеры IBM профессионального класса и пытаться обучать учащихся работе с современным профессиональным программным обеспечением? По мнению авторов, не следует. Постараемся обосновать, почему это так, с помощью четырех утверждений.

**Утверждение первое:** не следует путать обучение и передачу информации (инструктаж), хотя внешне эти процес-

сы могут показаться похожими. Обучение предполагает, что каждому из учащихся в ходе прохождения учебного курса придется проделать большую интеллектуальную работу, которая приведет к значительному развитию их мышления, памяти и творческих способностей. Передача же информации (инструктаж) не предполагает развития личности тех, кому эта информация передается. Как правило, такой инструктаж может быть сведен к простейшей передаче информации, например в печатном виде, и выдаче рекомендаций по обращению с ней. Возрастные особенности учащихся средних учебных заведений таковы, что все они нуждаются в первую очередь в развивающем обучении. Обрушивать на них лавину информации, к восприятию и использованию которой они ни интеллектуально, ни профессионально не готовы, — занятие бесполезное и даже вредное. Все это относится к азам педагогической науки. Поэтому обучение должно стимулировать напряженную умственную и духовную работу учащихся, и главный его результат — выработка у них способностей и навыков к такой работе. Этот процесс можно эффективно организовать при изучении основ алгоритмизации и программирования. Выработка умений построить, сообразить, проследить сложный вычислительный процесс означает выработку навыков динамического мышления, что является огромным шагом вперед в интеллектуальном развитии учащихся. Отсюда следует, что работа с какой-нибудь простой и наглядной системой программирования должна стать одним из основных средств развивающего обучения. Но должна ли это быть профессиональная система программирования?

Ответ на этот вопрос содержит в себе следующее утверждение.

**Утверждение второе:** профессиональные программные системы очень неудобны для первоначального обучения школьников. Это доказано опытом. Профессиональные программные системы (типа Turbo Pascal, FoxPro, C++ и т. д.) содержат огромное число не существенных для начального их понимания деталей, режимов, меню, сообщений. Особенно сложны для учащихся длинные англоязычные сообщения. Главная же трудность при использовании таких систем в качестве учебных состоит в том, что они рассчитаны на человека с уже подготовленным мышлением программиста-профессионала. Такого мышления у учеников на начальном этапе обучения просто нет. Следовательно, необходимы специализированные учебные программные системы. Насколько они должны быть похожи на те, которые в настоящее время используют профессиональные пользователи современных ИВМ? Ответ на этот вопрос содержит в себе следующее утверждение.

**Утверждение третье:** попытки учителя угнаться за техническим прогрессом и модой в области компьютерной техники все равно ни к чему не приведут. Любая информация о новейших технических и программных средствах безнадежно устареет, прежде чем учащиеся закончат обучение. Вывод: следует обучать лишь общим принципам работы с программными и техническими системами, которые не устареют в обозримом будущем. И главным результатом обучения должна быть не сумма переданных знаний (т. е. информации), а развитие мышления учащихся. Если этого удастся добиться, то их будущий переход к профессиональным программным системам может вообще не потребовать обучения, будет достаточно простой передачи информации, т. е. обычной инструкции (см. утверждение 1). Но можно ли добиться такого результата, используя лишь класс УКНЦ? Ответ на этот вопрос содержит следующее утверждение.

**Утверждение четвертое:** практика показывает, что конечные результаты обучения в терминальных классах зависят от применяемой программы, методики, контингента учащихся и практически не зависят от применяемого типа техники. Это подтверждается данными многочисленных проверок знаний, ре-

зультатами олимпиад, выпускных и приемных экзаменов.

Итак, получается, что для успешного обучения учащихся в терминальных классах, независимо от типа техники, нужны прежде всего специализированные учебные программные системы. Естественно, эти системы должны быть программно-методическими, т. е. включать в себя и необходимую методическую информацию для учителя, и, возможно, даже его обучение.

Сразу возникают вопросы: какими должны быть учебные системы? Существуют ли они для классов УКНЦ? Где их взять? Попытаемся ответить.

## 2. Какой должна быть учебная программно-методическая система

В учебную программно-методическую систему должна входить мощная и простая в обращении программная система, с помощью которой учащиеся могли бы создавать некоторые весьма сложные и в то же время наглядные и понятные динамические объекты (программы, алгоритмы, мультфильмы, тесты и т. д.). Этому требованию лучше всего удовлетворяют системы программирования с наглядным интерфейсом, т. е. системы, содержащие внутри себя некоторый язык программирования и средства создания и отладки программ. Разработка учебно-методической системы на основе какой-нибудь иной программной системы, например мощного графического редактора с возможностью мультипликации, представляется авторам весьма сомнительной, так как он значительно меньше приспособлен для конструирования и отладки динамических процессов.

Итак, учебная программно-методическая система должна включать в себя по крайней мере одну учебную систему программирования с наглядным интерфейсом. Ясно, что такая система должна обладать большими графическими возможностями. Кроме того, программная система должна быть упрощенной учебной моделью известных профессиональных систем программирования, т. е. содержать в себе все их существенные (и только существенные) элементы. Список же известных профессиональных систем программирования, которые реально



могут быть взяты в качестве прообразов, весьма невелик. Это Logo, Turbo Prolog, Turbo Pascal, Turbo Basic, и Turbo C для IBM. В последних трех системах важной деталью является наглядный интерфейс, т. е. наличие в этих системах встроенного интегрированного отладчика, позволяющего просмотреть на экране во всех деталях и во всей его динамике ход выполнения программы, т. е. процесс изменения переменных, переходы и процесс ввода-вывода. Общеизвестно, что работа с такой системой многократно повышает скорость создания и отладки программ (по заявлениям некоторых программистов, до 10 раз). Наглядный интерфейс стал в наше время фактическим стандартом, на котором основаны все вновь создаваемые системы программирования. Поэтому просто необходимо, чтобы этот принцип был заложен и в современную учебную систему программирования. Кроме того, работа с такой системой может помочь выработать у учащихся навыки динамического мышления, что является одной из главных задач обучения.

### 3. Какие существуют учебные программные системы

Исторически первой учебной программной системой была система Logo, созданная в 70-х годах С. Пейпертом в США. Эта система была основана на управлении движением объекта (так называемые черепашки) на экране. Позднее система была обогащена многими элементами языка LISP. Различные ее реализации быстро завоевали популярность в школах США и Западной Европы как средство развивающего обучения в средней и младшей возрастных группах.

Следующей серьезной учебной программной системой была система Е-практикум, разработанная в начале 80-х годов академиком А. П. Ершовым. Заложенный в ней язык программирования (так называемый «школьный алгоритмический язык») является фактически упрощенной русскоязычной версией языка Паскаль. Из наглядного интерфейса в ранних реализациях этой системы использовался лишь просмотр результатов операторов присваивания и пошаговый режим выполнения (см. системы КуМир разработки фирмы «ИнфоМир», Москва). Отношение учителей к этой системе было весьма неоднознач-

ным. Многие учителя восприняли эту систему лишь с технической стороны, т. е. как еще один алгоритмический язык, совершенно упустив из виду ее основное учебно-развивающее назначение. В результате большинство из них стали стихийно использовать в качестве учебных систем различные реализации языка Бейсик, которые создавались не для этой цели. Реализации языка Бейсик, которые реально использовались, не имели ни необходимого для учебной программной системы наглядного интерфейса, ни стандартизации с профессиональными системами (так называемые структурные элементы). В результате эффективность такого обучения оказалась низкой. Теперь же, по мере накопления опыта, многие учителя понимают, что Е-практикум заслуживает более серьезного отношения. Учащиеся, прошедшие обучение на этой системе, потом легко и просто осваивают Бейсик, Паскаль и любой другой язык. В данном случае можно сказать, что они уже прошли этап развивающего обучения и для освоения нового языка им нужно не обучение, а простой инструктаж (см. утверждение первое).

В учебнике информатики В. А. Каймина была сделана попытка использовать в качестве учебной программной системы различные программные реализации языка Пролог. Время не подтвердило плодотворность такого подхода. Видимо, потому, что выполнение программы на языке Пролог бывает трудно представить в виде наглядного динамического процесса, что затрудняет его учебное использование.

Все эти обстоятельства и накопленный опыт были использованы фирмой «Альтернатива» (г. Барнаул, до середины 1995 г. называлась «Новая альтернатива») при создании нового семейства учебных программных систем.

В 1994—1995 годах было создано следующее семейство учебных программных систем для классов УКНЦ:

БЕТА-ЛОГО,  
АЛТ-ПАСКАЛЬ,  
АЛТ-БЕЙСИК,  
АЛТ-Е-ПРАКТИКУМ,  
АЛТ-СИ.

Все эти учебные программные системы составят основу нового программно-методического пакета АЛЬТЕРНАТИВА-3 для классов УКНЦ.

Такое многообразие учебных систем

| Alt-Pascal   |        |   |
|--|--------|---|
| Процедура  | Курсор | Выполнить Сброс Значения Окна   |
| ввод-вывод   |        | просмотр  |
| Сумма= 7.50  |        | c:'0'<br>x1:20.5<br>y1:25.125<br>s:7.5<br>a[1]:0.5<br>a[2]:1<br>a[3]:1.5<br>a[4]:2.<br>a[5]:2.5 |
| 20 : 1 prim.pas  |        |   |
| <pre> Var  c      :Char;      x1,y1 :real;      x      :array[1..4] of real;  Begin   c:=chr(48); x1=20.5; y1=25.125p; s:=0;   for i:=to 4 do begin a[i]=0.5*i; s:=s+a[i]; end;   goto(2,5); writeln('Сумма=',s:6:2);   Ellips(x1,y1,s); End. </pre> |        |   |

Рис. 1

позволит каждому учителю выбрать для себя ту из них, которая ему больше подходит. Так, система БЕТА-ЛОГО может быть полезна при работе с младшими классами. Язык Паскаль считается наилучшим средством обучения программированию. Системы АЛТ-БЕЙСИК и АЛТ-Е-практикум будут полезны учителям, привыкшим к преподаванию этих языков. Систему АЛТ-СИ можно применять для факультативного изучения со способными учащимися. Каждая из этих систем содержит в себе текстовый редактор, интерпретатор соответствующего языка программирования и встроенный интегрированный отладчик, обеспечивающий работу в режиме «наглядного интерфейса». Все эти системы являются учебными моделями — аналогами соответствующих систем на IBM. При этом БЕТА-ЛОГО — это аналог LogoWriter, АЛТ-ПАСКАЛЬ — аналог Turbo Pascal v.5, АЛТ-БЕЙСИК — аналог QBasic, и АЛТ-СИ — аналог системы Turbo C v.2. Все эти системы стандартизированы между собой по текстовому редактору и по «наглядному интерфейсу», что сильно облегчает их совместное или последовательное изучение.

Поясним подробнее, что представляет собой «наглядный интерфейс» вышеперечисленных систем. На рис. 1 приведен пример экрана ученической машины УКНЦ при работе с небольшой учебной программой. Верхняя строка содер-

жит меню команд системы, доступных в режиме отладки программы. Экран УКНЦ разделен на три окна — ввода-вывода, текста программы и просмотра переменных. В окне ввода-вывода можно просмотреть текущие значения переменных и осуществляемый программой ввод-вывод. В окне текста можно просматривать часть текста отлаживаемой программы. Одна из строк в этом окне всегда выделена инверсным цветом, т. е. считается подлежащей выполнению. Если дать команду «Шаг», эта строка программы будет выполнена и, если нужно, изменится содержимое в окне переменных и окне ввода-вывода. В этом же окне всегда присутствует текстовый курсор. С его помощью можно задавать фрагмент программы (от выделенной строки до текстового курсора), подлежащий быстрому выполнению. Если дать команду «Курсор», то этот программный фрагмент будет быстро выполнен, и опять же может измениться содержимое окна ввода-вывода и окна просмотра переменных. Команда «Выполнить» приведет к быстрому выполнению программы от выделенной строки до конца.

Так можно последовательно выполнять программу по строкам, следя при этом за процессом изменения переменных. Менее важные фрагменты программы можно выполнять в быстром режиме, а более важные — в пошаговом. Команда «Значения» позволяет в любой момент

отредактировать содержимое окна просмотра переменных, удалив или вставив в него любую переменную, и, если нужно, принудительно изменить значение любой из них. Такая возможность детально проследивать ход вычислительного процесса является главным методическим достоинством учебных программных систем с наглядным интерфейсом, так как облегчает выработку у учащихся навыков динамического мышления.

#### 4. Заключение

Из всего сказанного выше следует, что при должном оснащении учебными программно-методическими системами класс УКНЦ вполне пригоден для эффективного обучения учащихся основам информатики и вычислительной техники. Остается лишь порекомендовать

всем средним учебным заведениям иметь у себя вышеназванные системы. Адреса их разработчиков многократно публиковались в журнале «Информатика и образование» в течение прошедшего года.

При этом, еще раз подчеркнем, не следует путать между собой некоторые внешне похожие процессы. Не следует путать обучение учащихся и их информирование о технических и программных новинках. Это совершенно разные по своей необходимости и целесообразности процессы. Не следует также путать развивающую ум и динамическое мышление работу с учебной программной системой, которая будет полезна всем учащимся, и профессиональную подготовку будущего программиста, которая нужна далеко не каждому.

#### НАУЧНО-ТЕХНИЧЕСКОЕ ПРЕДПРИЯТИЕ

### « *НОВАЯ АЛЬТЕРНАТИВА* »

**предлагает новейшие учебные программные системы для классов УКНЦ:**

**БЕТА-ЛОГО** — новая программная реализация языка Лого для обучения младших школьников. Является полной копией известной системы LogoWriter для IBM.

**АЛТ-ПАСКАЛЬ** — первая полная реализация системы Turbo Pascal для УКНЦ.

**АЛТ-Е-ПРАКТИКУМ** — дальнейшее развитие идей академика А. П. Ершова о школьном алгоритмическом языке.

**АЛТ-БЕЙСИК** — первая реализация системы Turbo Basic для УКНЦ объединяет в себе все достоинства языков Бейсик и Паскаль.

**АЛТ-СИ** — первая реализация системы Turbo C для УКНЦ.

Все АЛТ-системы имеют встроенную графику и интегрированный отладчик, позволяют детально проследивать ход выполнения любой программы и быстро находить и устранять в ней ошибки. Такой дружелюбный интерфейс уже стал международным стандартом для систем программирования.

Мы готовы поставить программные пакеты АЛЬТЕРНАТИВА для классов УКНЦ, КУВТ-86, «Корвет» и БК-0011. Состав пакетов — полный набор программного и методического обеспечения для учебного процесса (сетевые системы, текстовые, графические, музыкальные редакторы, базы данных, языки высокого уровня, техническая и методическая документация).

**Надежность нашей фирмы доказана восемью годами работы  
и наилучшими отзывами учителей.  
Наши заказчики есть повсюду.**

**Наш адрес: 656057, г. Барнаул, в/я 2513. НОВАЯ АЛЬТЕРНАТИВА.**

**Директор А. Н. Гриценко.**

# КЛУБ УКНЦ

**В. М. Карнаухов,**

*кандидат физико-математических наук, доцент кафедры высшей математики  
МСХА им. К. А. Тимирязева*

## СИСТЕМА КОНТРОЛЯ ЗНАНИЙ

### Общая характеристика системы

Предлагаемая система («Мудромер») позволяет автоматизировать проведение контрольных работ (КР) по любым дисциплинам в классах УКНЦ.

Система «Мудромер», созданная и апробированная в Тимирязевской академии в 1991—1993 гг., достаточно объективно оценивает знания учащихся. Как показал опыт внедрения в ТСХА, преподаватель может полностью отказаться от проведения письменных контрольных работ: оценки, получаемые студентами на письменных контрольных работах, выше оценок, выставляемых системой.

Связь преподавателей с системой происходит через оператора класса, который непосредственно работает с «Мудромером». Для проведения контрольной работы, задачи которой уже имеются в библиотеке вопросов, преподаватель заполняет лишь требование. Если контрольная работа включает задачи, которых нет в библиотеке, преподаватель передает оператору варианты новых задач для внесения их в систему. В конце компьютерной контрольной работы преподаватель получает удобный листинг с результатами опроса.

Требование преподавателя на проведение контрольной работы включает в себя (см. табл. 1): фамилию преподавателя, время проведения, длительность опроса, номер группы (класса); список группы (класса), необходимый в случае, если компьютерный контроль для данной группы проводится впервые, структуру КР, включающую в себя шифры каталогов библиотеки, номера задач этих каталогов и их весовые коэффициенты, выражающие уровень значимости задач.

Таблица 1

| Петров А. Г. | KWD |   | POD |   |   |   |
|--------------|-----|---|-----|---|---|---|
| 27.05.94     | 5   | 3 | 4   | 2 | 1 | 7 |
| 9.00—9.45    | 5   | 5 | 5   | 3 | 4 | 3 |
| 45'          |     |   |     |   |   |   |

Комплекс позволяет проводить одновременно или последовательно друг за другом до 10 контрольных работ. Загрузка системы «Мудромер» по сети на 12 РМУ при любой конфигурации класса происходит за 3—4 мин. Перезагрузка для проведения следующей КР не требуется. Достаточно, не покидая программы опроса, указать номер следующей контрольной работы.

Во время контрольной работы за одним компьютером могут опрашиваться одновременно двое учащихся. Тексты задач постоянно находятся на экране дисплея. Таким образом, в обычном классе УКНЦ (12 РМУ) система «Мудромер» позволяет проводить полноценные КР.

Система поставляется с библиотекой задач по математике для XI класса, составленных на основе дидактических материалов по алгебре (авторы Б. М. Ивлев, С. М. Саакян, С. И. Шварцбург) и по геометрии (авторы С. Б. Веселовский и В. Д. Рябчинская).

### Какие вопросы можно задавать учащимся

Система позволяет создавать для тестирования вопросы трех типов: логический, произвольный и числовой. Каждый тип определяется способом обработки ответа к задаче. Текст вопроса произволен и имеет формат:

18 строк × 38 столбцов.

Ответ к вопросам первого типа представляет собой набор любых символов, правильность каждого из которых зависит от правильности предыдущих. Такой ответ оценивается по доле правильно введенных символов до первого неправильного.

Пример 1 (тип 1).

Тригонометрическое выражение разложено на множители.  
 Ввести пропущенные символы в порядке их следования (точка соответствует одному символу).  

$$\frac{\cos^2 x + \cos^2 2x + \cos^2 3x + \cos^2 4x - 2 - \cos 2x \cdot \cos x + \cos 6x \cdot \cos 8x}{2} - \cos 3x \cos x \cdot \cos x - \cos x \cos x \cos 2x$$
  
 Примерный вид ответа: + - \* 2 1 -

Истинный ответ: + 4 + + 7 2 5

Текст в рамке заполняется оператором в произвольной форме при помощи встроенного текстового редактора. Пробелы при вводе ответа игнорируются системой при его анализе. Тип ответа определяется преподавателем. В данной ситуации можно считать, что зависимость между символами такая, что любой из них можно считать правильным в том и только в том случае, если все символы, введенные до него, и сам символ — правильные.

Приведем еще примеры вопросов логического типа.

Пример 2 (тип 1).

Ввести номера пропущенных в определении фрагментов (точка соответствует один фрагмент).  
 Теорема Пифагора:  
 . квадратов . равна . .  
 Фрагменты:  
 1. разность 2. сумма 3. сторон  
 4. произведение 5. кубу  
 6. квадрату 7. стороны  
 8. катетов 9. гипотенузы  
 А. биссектрисы В. медианы  
 Примерный вид ответа: 4125

Истинный ответ: 2869

Пример 3 (тип 1).

Расположить данные металлы по убывающей восстановительной способности:  
 1. Fe 2. Ca 3. Pb 4. Al 5. Cu  
 В ответе указать номера металлов в порядке убывания.  
 Примерный вид ответа: 42315

Истинный ответ: 24135

Пример 4 (тип 1).

Указать пропущенные символы в порядке их следования (точка соответствует один символ):  
 $\text{Na}_2\text{O} + \text{H}_2\text{O} - \dots$   
 Образец ответа: 3HNaO

Истинный ответ: 2NaOH

Пример 5 (тип 1).

Используя слова:  
 1. English 2. you 3. do 4. speak,  
 спросить по-английски: «Вы говорите по-английски?»  
 В ответе указать номера слов в порядке их следования в вопросе.  
 Примерный вид ответа: 4321

Истинный ответ: 3241

Пример 6 (тип 1).

Имеется программа нахождения квадратного корня из числа. Ввести пропущенные символы в программе (точка соответствует один символ).  

```

10 INPUT X
20 IF X>=0 THEN .0
.0 PRINT SQR(.)
.0 GOTO .0
.0 PRINT "Корень вычислить нельзя"
60 END
    
```

  
 Примерный вид ответа: 3 X 4 3 2 5

Истинный ответ: 5 5 X 4 6 3

## Пример 7 (тип 1).

Построить общую схему решения задач при помощи ЭВМ, используя следующие блоки:

1. Алгоритм
2. Математическая модель
3. Программа на ЭВМ
4. Реальная задача
5. Анализ результатов работы ЭВМ

В ответе указать номера блоков в порядке их использования при решении задачи.

Примерный вид ответа: 43125

Истинный ответ: 42135

Ответ к вопросам второго типа представляет собой набор любых независимых друг от друга символов. Правильность такого ответа оценивается по доле правильно введенных символов. Ниже приводятся примеры вопросов такого типа.

## Пример 8 (тип 2).

Ввести пропущенные символы в формуле для корней квадратного уравнения (точке соответствует один символ).

$$ax^2 + 2bx + c = 0$$

$$x = \dots \pm \sqrt{\dots} \dots \times c$$

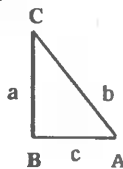
Примерный вид ответа: + - 4 2 a

Истинный ответ: - b b a a

## Пример 9 (тип 2).

Ввести номера пропущенных фрагментов в предложенных формулах (точке соответствует один фрагмент):

- a = b × . (A)  
 c = a × . (A)  
 c = a × . (C)  
 b = c / . (C)



Фрагменты:

1. cos 2. sin 3. tg 4. ctg

Примерный вид ответа: 1234

Истинный ответ: 2431

## Пример 10 (тип 2).

Какие из нижеприведенных утверждений верны (Y), а какие нет (N)?

- Гидролизу подвергается соль  
 1. KCl 2. Na<sub>2</sub>SO<sub>4</sub> 3. ZnCl<sub>2</sub>  
 4. Ba(NO<sub>3</sub>)<sub>2</sub>

Примерный вид ответа: Y N Y N

Истинный ответ: NNYN

## Пример 11 (тип 2).

Ввести пропущенные символы (точке соответствует один символ, отсутствие символа обозначается &):

In the f.rst sch.ol year they st.died elements of hig.er algebra, anal.tic geometr., set theor. and the c.lculus of f.nction of one real v.riables

Примерный вид ответа: abwpral&ne

Истинный ответ: iouhyuaua

К третьему типу относятся задачи с числовым либо векторно-числовым ответом (компоненты вектора вводятся через точку с запятой). Причем числа либо компоненты вектора могут быть записаны при помощи формул с использованием всех арифметических операций и стандартных функций.

## Пример 12 (тип 3).

Вычислить значение интеграла:

$$\int_0^1 x^2 dx$$

Истинный ответ: - 1/3

## Пример 13 (тип 3).

Функция  
 $F(x) = x/2 - 2/x$   
 является первообразной функции  
 $f(x) = 1/m + n/x^k$ ,

Найти m, n, k и ввести их через точку с запятой в указанной последовательности.

Истинный ответ: - 2;2;2

Пример 14 (тип 3).

Методом интервалов решить неравенство:

$$\frac{\log_{0.4}(x-2)}{x-4} \leq 0$$

В ответе указать конечные границы множества решений по возрастанию через точку с запятой.

Примеры ответов:

Множество решений:  $(-3;5)+(23;\infty)$

Ответ: -3;5;23

Множество решений:  $(45;\infty)$

Ответ: 45

Множество решений:  $(2;9)+(9;33)$

Ответ: 2;9;9;33

Истинный ответ: 2;3;4

Найти координаты вектора  $\bar{a}$ , если  $\bar{a} = 3\bar{b} - 4\bar{c}$ ,  $\bar{b} = (1; 1)$ ,  $\bar{c} = (0; -3)$ .

Координаты векторов  $\bar{x}$  и  $\bar{y}$  ввести через точку с запятой в указанной последовательности.

Примерный вид ответа: 4/5;-2

Истинный ответ: 3;15

Пример 15 (тип 3).

Произвольность текста вопроса позволяет составлять различные вопросы с малой вероятностью простого угадывания правильного ответа. Сравните:

Пример 16 (тип 3).

Какое из множеств не содержит отрицательных чисел:

- 1) множество целых чисел,
- 2) множество натуральных чисел,
- 3) множество рациональных чисел,
- 4) множество иррациональных чисел,
- 5) множество действительных чисел.

В ответе укажите номер множества.

Истинный ответ: 2

Пример 17 (тип 3).

Верны (Y) или неверны (N) данные утверждения?

Отрицательные числа содержатся во множестве:

- 1) действительных чисел,
- 2) рациональных чисел,
- 3) иррациональных чисел,
- 4) натуральных чисел,
- 5) целых чисел.

Примерный вид ответа: YNYNY

Истинный ответ: YYYNY

Вероятность простого угадывания правильного ответа в примере 16 равна  $1/5$ , в примере 17 —  $1/32$ .

**Файловая структура библиотеки вопросов**

Библиотека вопросов разделяется на каталоги. Каждому каталогу соответствуют на диске файлы с определенным расширением, которое задается оператором в виде шифра каталога. Например, каталог по теме «Квадратные уравнения» (шифр KWD) может содержать файлы

F02.KWD, F05.KWD, FILE.KWD, F03.KWD, F01.KWD, F04.KWD, ....

Файл типа FNN.KWD ( $00 < NN < 41$ ) содержит варианты одной задачи по теме «Квадратные уравнения». Число вариантов задачи определяет преподаватель и не может превосходить 50. Система случайного распределения вариантов такова, что даже при небольшом их количестве позволяет избежать ситуации, когда за одним компьютером учащимся решается одна и та же задача.

Файл FILE.KWD создается системой и содержит краткий перечень всех задач данного каталога и номеров файлов, в которых находятся варианты задач.

Библиотека размещается на одном физическом или логическом устройстве.

Такая структура библиотеки позволяет:

- а) без особых усилий перемещать каталоги с одного устройства на другое,
- б) оперативно искать ту или иную задачу в библиотеке вопросов,
- в) достаточно гибко решать проблему составления равноценных билетов для опроса учащихся.

## Связь системы с преподавателями

Одним из недостатков современных прикладных программных средств является то, что они обычно рассчитаны на узкий круг пользователей, от которых требуется по крайней мере сесть за компьютер и разобраться с работой этих программ.

Предлагаемая система рассчитана на преподавателя, от которого требуется лишь знание своего предмета. Связь с системой осуществляется посредством документации (см. ниже), позволяющей преподавателю отслеживать состояние библиотеки вопросов, и оператора, который непосредственно работает с компьютером.

Таблица 2

**Сетка-календарь заявок преподавателей  
на проведение компьютерного тестирования**

| Время     | 1 | 2   | 3 | 4 | Суб | Вс | 7   | 8 | 9 | 10 | 11  | Суб | Вс | 14 | 15 | 16 |
|-----------|---|-----|---|---|-----|----|-----|---|---|----|-----|-----|----|----|----|----|
| 8.00—8.45 |   | Пет |   |   |     |    | Кар |   |   |    | Пет |     |    |    |    |    |
| 8.50—9.35 |   | Пет |   |   |     |    | Кар |   |   |    | Пет |     |    |    |    |    |

В пакет документов, которые можно получить на принтере, входит

- сетка-календарь для отметок преподавателей о проведении контрольной работы в определенный день и час (см. табл. 2);
- лист для заполнения требования с указанием структуры билета (см. табл. 1);
- сжатая и расширенная формы каталогов библиотеки вопросов (сжатая форма представляет собой содержимое файла FILE.\*, т. е. информацию о названиях задач и номерах файлов FNN.\*, в которых содержатся варианты этих задач, а расширенная — распечатку первых вариантов задач, упомянутых в сжатой форме, см. табл. 3).

Таблица 3

### Сжатый каталог EXA

Каталог EXA

Примерные задачи по школьным предметам.

1. Сложение двух целых чисел.
2. Упрощение выражения.
3. Уравнение химической реакции.
4. Программа на языке Бейсик.

### Расширенный каталог EXA

1. Сложение двух целых чисел.  
Найти сумму  $1342 + 2768$ .

В ответе указать значащие цифры полученного результата в порядке их следования в сумме через точку с запятой.

2. Упрощение выражений.

Упростить выражение:

$$\frac{a^2 - b^2}{a - b}$$

В ответе ввести значение упрощенного выражения при  $a - b = 2$ .

3. Уравнение химической реакции.

Написать уравнение химической реакции взаимодействия калия с водой. В ответе ввести пропущенные в предложенном уравнении коэффициенты в порядке их следования (точке соответствует один коэффициент).



Примерный вид ответа: 1234

4. Программа на языке Бейсик.

Предлагаемая программа должна вычислять сумму квадратов чисел одномерного массива:

```

10 S=
20 FOR I=1 TO 10
30 INPUT A(I)
40 NEXT I
50 FOR I=1 TO 10
60 .-S.A(I)*.(.)
70 NEXT I

```

В ответе ввести пропущенные символы в порядке их следования в тексте (точке соответствует один символ).

Примерный вид ответа: S A I \* +

*Замечание.* Первая и вторая задачи сформулированы таким образом, чтобы



ученик не смог воспользоваться анализатором формул.

Примеры набора арифметических формул, играющего роль HELP-листа для учащихся, см. в табл. 4.

Таблица 4

**Примеры набора арифметических формул**

Арифметическое выражение:

$$34 \times 79$$

$$45 \times 65$$

Пример ввода:

$$34*79/45/65 \text{ или } 34*79/(45*65)$$

Арифметическое выражение:

$$2e$$

$$\pi$$

Пример ввода:

$$2*EXP(1)/PI$$

- итоговый листинг с результатами опроса (см. табл. 5).

Таблица 5

**Итоговый листинг результатов опроса**

Контроль по теме «Теоремы сложения и умножения» (12 агро.; 23.03.94).

Длительность опроса: 90.

Структура билета.

PRB1. Задачи на теоремы сложения и умножения вероятностей.

Вес задачи — 5.

PRB6. Задачи на формулу полной вероятности.

Вес задачи — 5.

VER16. Формула Бернулли.

Вес задачи — 5.

Максимальный рейтинг: 35

| Фамилии   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | S    | O   |
|-----------|---|---|---|---|---|---|---|------|-----|
| Арцимович | - | 5 | - | 5 | 5 | 5 | 5 | 25.0 | 3.6 |
| Власов    | 0 | 5 | 5 | 5 | 5 | 5 | 3 | 28.0 | 4.0 |
| Дьякова   | - | - | 3 | - | 4 | 0 | 5 | 12.0 | 1.7 |

Средний балл группы: 2.8

**Особенности программ системы**

Общение оператора с системой «Мудромер» происходит через управляющее меню:

- УПР/V — подготовка вопросов;
- УПР/L — подготовка документации;
- УПР/H — подготовка к опросу;
- УПР/K — опрос учащихся;
- УПР/A — анализ результатов.

Каждая программа системы, выполняющая одну из вышеперечисленных функций, представляет собой текстовый редактор, при помощи которого осуществляется диалог между пользователем и «Мудромером».

Окно редактора вопросов постоянно содержит информацию о шифре каталога, номерах задачи и набираемого варианта, типе вопроса, числе вариантов задачи.

При работе с редактором постоянно отображается HELP-справка о его возможностях, среди которых можно отметить операцию запоминания текста, позволяющую быстро размножить варианты задачи.

При подготовке документации можно не только распечатать необходимую для массового пользователя документацию, но и:

- отредактировать названия каталогов, задач;
- удалить задачу из каталога;
- копировать на архивные дискеты задачи и каталоги.

Программа подготовки к опросу служит для задания параметров опроса, которые указал преподаватель в своей заявке на проведение контрольной работы. Отметим, что помимо длительности опроса, шифров каталогов, номеров задач и их весовых коэффициентов оператор должен задать ряд параметров.

1. Число приоритетных тем K, указывающее на число первых задач опроса, которые учащиеся должны обязательно решить и набрать больше половины возможных баллов для того, чтобы получить доступ к другим задачам. Таким образом, система позволяет проводить опросы двух типов: контрольную работу с равным доступом учащихся ко всем задачам (K=0) и экзамен с обязательным решением первых задач (K>0).

2. Информационную строку о характере и дате проведения опроса. Эта строка войдет в итоговый листинг результатов опроса.

3. Имя файла, в котором хранится либо будет храниться список группы учащихся.

4. Номер контрольной работы (заранее можно подготовить 10 контрольных работ).

После задания структуры билета и списка учащихся необходимо (!) сформировать билеты, так как во время опроса эту операцию выполнить невозможно

при одновременном обращении всех РМУ к РМП.

После задания параметров можно приступать непосредственно к опросу учащихся. Загрузка программы опроса на 12 РМУ происходит за 3—4 минуты при любой конфигурации класса (отсчет времени ведется с момента включения рубильника). После появления заставки оператор или учащиеся указывают номер контрольной работы и вводят свои фамилии. Если за компьютером работают два учащихся, то фамилии набираются ими последовательно. При работе двух учащихся за РМУ экран дисплея делится пополам разделительной чертой, ответы вводятся поочередно. При этом слева сидящий ученик перед ответом нажимает <1>, справа сидящий — <2>. Отметим, что для ответа на вопрос с числовым ответом учащемуся предоставляется три попытки для того, чтобы исключить его неточности в промежуточных вычислениях. Каждый учащийся имеет право перейти к следующему вопросу, перейти сразу к вопросу с заданным номером, отказаться от ввода ответа и передать это право своему соседу, оценить оставшееся время для решения задач, досрочно получить итоговую оценку и завершить работу с программой опроса. Итоговая оценка представляется в двух видах: в виде рейтинговой оценки и уточненной пятибалльной оценки с одним знаком после запятой.

После приглашения оператора последний сбрасывает результаты опроса на РМП. Такая концовка опроса не позволяет терять данные.

При работе одного учащегося за компьютером в любой момент к нему может подключиться другой учащийся.

Отметим, что связь РМУ с РМП происходит лишь в начале и конце опроса, что существенно разгружает сеть и позволяет оператору во время опроса отключать РМП от сети.

После опроса вместе с оператором преподаватель может:

1) устранить такие недоразумения, как:

- учащийся решил задачу верно, а программа из-за ошибки в библио-

теке вопросов выставила нулевую оценку;

- не хватило времени для ввода уже готового ответа.

2) распечатать листинг результатов опроса,

3) распечатать билет любого учащегося (либо для более детального анализа его ответов, либо до опроса в случае, когда число учащихся в группе больше рабочих мест в классе). В этой ситуации учащийся получает распечатку своего билета и решает задачи без компьютера. После освобождения одного рабочего места этот учащийся быстро может ввести полученные ответы и получить итоговую оценку.

### Результаты использования системы в учебном процессе

Система использовалась автором и преподавателями кафедры высшей математики Тимирязевской академии в течение трех учебных лет (1991—1994 учебные годы). Некоторые преподаватели полностью отказались от проведения письменных контрольных работ, заменили частично проведение зачетов и теоретических экзаменов. Другие — пользовались услугами «Мудромера» в «разумных пределах», заменяя компьютерным контролем слишком трудоемкие контрольные работы, отнимающие много сил и времени для проверки. За три года проведения компьютерных контрольных работ не было ни одного срыва.

Анкетирование студентов показало, что они на 90% поддерживают новую форму контроля, при которой они получают оценку:

а) независимо от мнения преподавателя;

б) сразу же в конце контрольной работы;

в) как в виде рейтинга, так и в виде уточненной пятибалльной оценки.

Против новой системы контроля высказывались лишь те студенты, для которых «Мудромер» стал непреодолимым барьером на пути к «заветной тройке».

# НАПЕЧАТАНО В 1995 ГОДУ

## 10 ЛЕТ ШКОЛЬНОЙ ИНФОРМАТИКЕ

- Гейн А. Г., Шолохович В. Ф. Десять лет спустя 2
- Ершов П., Звенигородский Г. А., Первин Ю. А. Школьная информатика (концепции, состояния, перспективы) 1
- Каймин В. А. Как все нечилось... 3
- Умаров А. Ю. Вступая в век информации 2
- Христочевский С. А. Компьютер и образование 3

## РЕГИОНАЛЬНАЯ ПРОГРАММА

- Коган Е. Я., Первин Ю. А. Курс «Информационная культура» — региональный компонент школьного образования 1

## МОСКОВСКИЙ ВЫПУСК

- Болотова Н. В., Деревягина М. В., Ржевский Е. Ф. Компьютер на уроках технологии (первый опыт и ближайшие перспективы) 5
- Гадюлина И. В., Константинова Т. Д., Пудиков А. Д., Репченко В. Б., Семенов А. А., Юдина П. А. Использование востественнонучных дисков CD-ROM в образовательном процессе 5
- Горбунов А. Л. Компьютерные сети для детей через Lego TC Logo 5
- Курнешова Л. Е. От инвастий в образование зависит будущее России 5
- Кутукова О. Г. Методика подведения итогов Малышева М. А. Дети и Лого 5
- Митина О. В. Программная среда Лого глазами психолога 5
- Нестеров В. М. Компьютерный мини-класс для слепых и слабовидящих детей 5
- Переверзев Л. Б. Полюбить машины, помогающие учиться (Образовательная философия Сеймура Пейперта) 5
- Семенов А. Л. Математическая информатика в школе 5
- Семенов А. Л. Образование, информатика, компьютеры 5
- Сопрунов С. Ф. Лого-среда для детского творчества 5
- Станченко Н. С. Фигуристы, клумбы и Чучело — образные ассоциации в ЛогоМирах 5
- Юдина А. Г. История с зайцем 5

## ОБЩИЕ ВОПРОСЫ

- Базлов И. Ф., Шляго А. Н. Экзаменационные материалы по информатике (К разработке образовательных стандартов Санкт-Петербургской школы) 2
- Белкин А. Р., Шумов С. И. Анализ и оценка традиционных и нетрадиционных механизмов получения и обобщения новых знаний 1
- Болотов В. А. Мы можем производить программное обеспечение на мировом уровне 4
- Демущкин А. С., Кириллов А. И., Сливина Н. А., Чубров Е. В., Кривоше-

- ев А. О., Фомин С. С. Компьютерные обучающие программы 3
- Информационные письма Министерств образования РФ 2, 4, 6
- Конкурс на разработку федеральных компонентов государственных образовательных стандартов 2
- Материалы коллегии Министерств образования РФ 4
- Первин Ю. А. Информационная культура и информатика: отдельно или слитно? 3
- Переверзев Л. Б. Полюбить машины, помогающие учиться (Образовательная философия Сеймура Пейперта) 6
- Рекомендации международной научно-практической конференции «организация единого информационного пространства средствами телекоммуникаций» (г. Пермь, 6–9 декабря 1994 г.) 2
- Самовольнова Л. Е. Комментарий к решению коллегии 4

## МЕТОДИКА

- Грамолин В. В. Хочу работать в Windows! (Уроки работы в текстовом редакторе Word for Windows 6.0 русская версия) 5
- Зайдельман Я. Н. Однопроходные алгоритмы 4
- Залогова Л. В., Русаков С. В., Семанкин И. Г., Шестакова Л. В. Базовый курс ОИВТ: «Пермская версия» 1, 2, 3, 5, 6
- Залогова Л. А., Семанкин И. Г. Уроки по Windows 4, 5, 6
- Исаков В. Н., Исакова В. В. Алгоритмизация и программирование: методические аспекты 2
- Кузьменко М. А. Как подготовить гипертекст 3
- Ладыженская Н. В., Зюкина З. С. Какой путь выбрать? (Гуманитарная информатика в школе) 3
- Маслов А. Н. Для современной школы есть выбор 1
- Пронина С. Е. Лого в школьном курсе информатики 2
- Радченко Н. П. От компьютерной грамотности — к культуре компьютерного века 3
- Радченко Н. П. Гуманизация курса ОИВТ 6
- Сорокин К. А. Музыкальные возможности Лого 6
- Сычев Н. А. Подготовка специалистов по информатике для науки и бизнеса 3
- Шафрин Ю. А. Чему учить школьников на уроках информатики? 5
- Шень А. Х. Программирование: теоремы и задачи 1
- Эпиктетов М. Г. НовоМир 6
- Юдина А. Г. Информатика на Лого для старшкклассников 1, 2

## ЗАДАЧИ

- Белова Л. Ю., Кузнецов В. А., Белов Ю. А., Епимахов С. В. Задача изображений сечений многогранников 6

- Варакин Ю. С.** Проекты-задания на языке Лого для учащихся старших классов 1, 3, 6  
1, 2
- Нечаев В. М.** Одна задача – три языка 1, 2
- Софронова Н. В.** Решение задач на массивы с помощью электронных таблиц 2
- Сычев Н. А.** Задания для вступительных экзаменов по информатике в НГУ 2

## ИНФОРМАТИКА В МЛАДШИХ КЛАССАХ

- Горячев А. В.** «Информатика в играх и задачах» 6
- Дубинина В. В.** Уроки развития, или Пропедевтический курс информатики для малышей 3
- Коляда Е. П.** Развитие логического и алгоритмического мышления учащихся I класса 6

## ПЕДАГОГИЧЕСКИЙ ОПЫТ

- Гольдшлаг О. Я.** Готовимся к выпускным экзаменам 6
- Дашниц Н. Л.** «Большие проекты» – в средней школе 3
- Мануйлов В. Г.** Основы информационных технологий (Программа прикладного курса для VI–XI классов с экономической подготовкой) 3
- Никольский А. Е.** Примерное поурочное планирование учебного материала для VI класса 3

## НОВЫЕ ПРОЕКТЫ

- Белич Д. В.** Сменим доску и мел на экран и лазер 5
- Богданова Д. А., Федосеев А. А.** Дистантное обучение: модерватор телеконференций 1
- Буданов В. М., Савицкий К. В.** Соревнования роботов во Франции 6
- Захарова Е. А.** Развивающий экран 6
- Козлов О. А., Солодова Е. А., Холодов Е. Н.** Некоторые аспекты создания и применения компьютеризированного учебника 3
- Ловцова Д. А., Сухов А. В.** Фрагмент компьютеризированного учебника для контроля знаний 3
- Казakov В. Г., Дорошкин А. А., Задорожный А. М., Князев Б. А.** Лекционная мультимедиа аудитория 4
- Пронин В. Н.** MultiVision Pro – год 1995 2
- Томенцева Е. Ю.** MultiVision и MultiVision Pro в учебном процессе 2
- Урнов В. А.** Вдоль по дороге... 5
- Утлинский Е. В., Авакумов Э. Г.** Телекоммуникационная сеть TIT-BIT 1

## APPLE ДЛЯ ОБРАЗОВАНИЯ

- Блинов А. Г.** Комплект русскоязычного программного обеспечения на CD-ROM 5
- Блинов А. Г.** Программы для Macintosh 5
- Дуайер Д.** Apple-классы завтрашнего дня: чему мы научились 3
- Конашевич М.** Технологии Apple в образовании 3
- Конашевич М.** Учиться играя 2
- Матвеева Т. А.** Электронный практикум по математике в средней школе-4 4
- Николайчук Г. С., Титова Ю. Ф.** Графиче-

ский редактор Kid Pix на уроках информатики 6

## ТОЧКА ЗРЕНИЯ

- Гриценко А. Н., Тушев А. Н.** Возможности и перспективы классов УКНЦ в учебном процессе 6

## ПРОФЕССИОНАЛЬНЫЙ КОМПЬЮТЕР

- Краснова О. А., Левченко И. В.** Излучение компьютера и здоровье детей 1
- Кузьменко М. А.** Персональные компьютеры вчера и сегодня 6
- Ловцов Д. А.** Защита информации 4
- Териновская Т. С.** Вопросы маркетинга программных продуктов 5
- Шумов С. И.** Телевизор моей мечты 6

## ЛЮДИ, ИДЕИ, РЕШЕНИЯ

- Частиков А. П.** История информатики в лицах: Грейс Мюррей Хоппер 3

## ЗАРУБЕЖНЫЙ ОПЫТ

- Виве М.** Образовательная инженерия 2
- Коски Л.** Обучение предпринимательской деятельности 2
- Карки С.-Л.** Современные тенденции в педагогике Финляндии 2
- Уинифрей Ф.** Мир через электронные послания 3
- Хирви В.** Роль образования в современной Финляндии 2

## КЛУБ БК

- Высоцкий И. Р.** Курс пользователя ПЭВМ на базе класса БК-001М 3

## КЛУБ УКНЦ

- Демаков А. В., Лебедев М. В.** Исполнители в системе КуМир 2
- Карнаухов В. М.** Система контроля знаний 6
- Моряхин А. Н.** Коррекция программы в ходе ее выполнения 3
- Файзуллин В. А.** Реализация алгоритма формирования массива с различными случайными элементами 1
- Фролов М. И.** Автоматизированная система построения расписания «Звуч» для УКНЦ 2

## КЛУБ «КОРВЕТ»

- Яромко Л. И.** Партитура в алгоритмических рамках 1

## ИНФОРМАЦИЯ

- Быстров И. И.** АО «ЛинТех» отвечает на вопросы 3

## НАМ ПИШУТ

- Розенфельд Д. И.** Три программы для Плюшкина 4
- Фадеев И. Ю.** Свет мой, зеркальце, скажи... 3
- Фадеев И. Ю.** Факторивл? Не просто, а очень просто 4
- Корхова В. П., Рыспаев Б. Б.** Телекоммуникационный проект «Пилотные школы» 4

# ИНФО

## 6'95

Научно-методический журнал  
Учрежден  
Министерством образования  
Российской Федерации  
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. А.  
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

### ОБЩИЕ ВОПРОСЫ

- Переверзев Л. Б. Полюбить машины, помогающие учиться  
(Образовательная философия Сеймура Пейперта) (продолжение) 2
- Информационные письма Министерства образования  
Российской Федерации 9

### МЕТОДИКА

- Залогова Л. В., Семакин И. Г. Уроки по Windows  
(продолжение) 13
- Залогова Л. В., Семакин И. Г. Базовый курс ОИВТ:  
«Пермская версия» (часть 8) 24
- Эпиктетов М. Г. НовоМир 41
- Радченко Н. П. Гуманизация курса ОИВТ 47
- Сорокин К. А. Музыкальные возможности Лого 51

### ПЕДАГОГИЧЕСКИЙ ОПЫТ

- Гольдшлаг О. Я. Готовимся к выпускным экзаменам 57

### НОВЫЕ ПРОЕКТЫ

- Буданов В. М., Савицкий К. В. Соревнования роботов  
во Франции 63
- Захарова Е. А. Развивающий экран 67

### ИНФОРМАТИКА В МЛАДШИХ КЛАССАХ

- Коляда Е. П. Развитие логического и алгоритмического  
мышления учащихся I класса 69
- Горячев А. В. «Информатика в играх и задачах» 79

### ЗАДАЧИ

- Варакин Ю. С. Проекты-задания на языке Лого для учащихся  
старших классов (продолжение) 81
- Белова Л. Ю., Кузнецов В. А., Белов Ю. А.,  
Епимахов С. В. Задача изображения сечений многогранников 86

### APPLE ДЛЯ ОБРАЗОВАНИЯ

- Николайчук Г. С., Титова Ю. Ф. Графический редактор KidPix  
на уроках информатики 89

### ПРОФЕССИОНАЛЬНЫЙ КОМПЬЮТЕР

- Кузьменко М. А. Персональные компьютеры вчера и сегодня 99
- Шумов С. И. Телевизор моей мечты 108

### ТОЧКА ЗРЕНИЯ

- Гриценко А. Н., Тушев А. Н. Возможности и перспективы  
классов УКНЦ в учебном процессе 113

### КЛУБ УКНЦ

- Карнаухов В. М. Система контроля знаний 118

### НАПЕЧАТАНО В 1995 ГОДУ

125

# CONTENTS

## GENERAL PROBLEMS

- Pereverzev L.** To Love the Machines Which Help Us to Learn (Educational Philosophy by Seymour Papert) (*continuation*) 2  
Information of the Ministry of Education of Russian Federation 9

## METHODICS

- Zalogova L., Semakin I.** Windows Lessons (*continuation*) 13  
**Zalogova L., Semakin I.** BCSCCT Basic Course: "Perm Version" (Part 8) 24  
**Epiktetov M.** NovoMir 41  
**Radchenko N.** Gumanization of the BCSCCT Course 47  
**Sorokin K.** Musical Opportunities of Logo 51

## PEDAGOGICAL EXPERIENCE

- Goldshlag O.** Preparing for Graduation Examinations 57

## NEW PROJECTS

- Budanov V., Savitskiy K.** Robots Competitions in France 63  
**Zakharova E.** Educational Screen 67

## INFORMATICS IN PRIMARY SCHOOL

- Kollada E.** Development of Logical and Algorithmical Thinking of the Pupils of the 1st Form 69  
**Goriachev A.** Informatics in Games and Tasks 79

## TASKS

- Varakin Yu.** Draft-assignments in Logo Language for Senior Pupils (*continuation*) 81  
**Belova L., Kuznetsov V., Belov Yu., Epimakhov S.** The problem of Polyhedron Cuttings Representation 86

## APPLE FOR EDUCATION

- Nicolaichuk G., Titova Yu.** Graphic Editor KidPix at Computer Lessons 89

## PROFESSIONAL COMPUTER

- Kuzmenko M.** Personal Computers Yesterday and Today 99  
**Shumov S.** TV of My Dream 108

## POINT OF VIEW

- Gritsenko A., Tushev A.** Possibilities and Perspectives of Computer UKNTs Courses in the Educational Process 113

## UKNTS CLUB

- Karnaikhov V.** Knowledge Control System 118

## PRINTED IN 1995

- 125
- На 1-й странице обложки — менеджер по маркетингу отдела образования фирмы Apple Computer С. И. С. Марина КОНАШЕВИЧ представляет новый компьютер Performa

## Директор

Кравцова А. Ю.

## Заместитель директора

Васильева Н. А.

## Ответственный секретарь

Иванова Т. В.

## Редакционная группа:

Иванова Т. В.,  
Кириченко И. Б.,  
Козырева Н. Ю.

## Компьютерная верстка

Кириченко И. Б.

## 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

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 03.11.95. Формат 70x100<sup>1/16</sup>. 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 10 050 экз. Заказ 1450.

Цена по подписке:  
для индивидуальных подписчиков — 8 000 руб. (индекс 70423);  
для предприятий и организаций — 25 000 руб. (индекс 73176).

Ордена Трудового Красного Знамени Чеховский полиграфический комбинат Комитета по печати и информации Российской Федерации. 142300, Чехов Московской обл.

© «Информатика и образование», 1995

Всероссийский конкурс «Учитель года России - 95» проходил в Москве с 27 сентября по 5 октября. В нем участвовали двое учителей информатики (фото 1): Н. П. Смирнова, учительница ОИВТ и логики 1-й Пензенской классической гимназии им. В. Г. Белинского и В. А. Усанов, учитель физики и информатики 301-й школы Москвы. Открытые уроки (фото 2, 3, 4) и доклады учителей информатики оценивались в сравнении с открытыми уроками и докладами учителей математики, так как предметное жюри было общим. Надеемся, что в будущем году число учителей информатики в финале конкурса увеличится. А мы совместно с отделом информатизации Министерства образования РФ будем добиваться организации отдельного предметного жюри по информатике.

**Участвуйте и побеждайте!**



Фото 1

Фото 2



Фото 3



Фото 4



# Virtual/PC

АО "ЛинТех"

119501 Москва, д/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" позволяют 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.**

