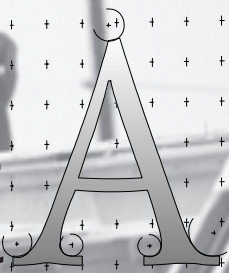


№ 4 (341) 23–31 января 2002

ИНФОРМАТИК



Еженедельная газета Издательского дома «ПЕРВОЕ СЕНТЯБРЯ»
ПОДПИСКА: (095) 249-47-58

Тропинка конструктора

А.А. Дуванов,
Ю.А. Первин

**Материалы
Роботландского
университета**

Книга для ученика

Вводная глава

В качестве вступления

Персонажи этой книги — бывший школьный учитель, а ныне пенсионер дедушка Фёрстов и его внуки, пятиклассники Дина и Тим, — хорошо знакомы слушателям Роботландского университета: на курсе “Зимние вечера. Информатика для начинающих” эта тройка выступала отдельной командой.

У дедушки есть, конечно, и имя, и отчество — его зовут Георгий Аркадьевич, но все его приятели и коллеги зовут его просто по фамилии — “дедушка Фёрстов”. Даже внуки его тоже так иногда называют. Правда, только за глаза, когда дедушки нет рядом.

Ребятам очень понравилось переписываться со своими сверстниками со всех концов России, участвовать в разнообразных конкурсах и турнирах, решать на компьютерах интересные задачи.

Поэтому в тот же день, когда со страниц уважаемой дедушкой Фёрстовым газеты “Информатика” им стало известно о новом наборе на курсы Роботландского университета, они уговорили дедушку стать руководителем команды, составленной из Дины и Тима. Впрочем, долго уговаривать не пришлось: Георгий Аркадьевич любит отдавать свое время и знания юным внукам. Он помог ребятам выбрать курс, который в большей мере, чем другие, мог бы быть продолжением “Зимних вечеров”. Так в Роботландском университете, на курсе “Компьютерное конструирование” вновь появилась команда Фёрстовых.

Знакома семья Фёрстовых и читателям “Информатики”: вечера, которые дедушка проводил с внуками, объясняя им решения задач, подробно описаны в газетных номерах первой половины 2001 года.

Дина и Тим, конечно, не такие знатоки информатики, как дедушка. Но их не смущают многие основные понятия начал этой науки, их лексикон богат заковыристыми терминами, а клавиатура и мышь для них так же привычны, как пульт управления телевизором.

Читатель, который знаком со школьным курсом “Роботландии”, будет находиться с Диной и Тимом в равном положении.

Для остальных предлагается краткое и порой не совсем строгое определение тех терминов и понятий, которые используют дедушка Фёрстов, Дина и Тим в своих диалогах, считая их для себя известными.

Роботландия

Предприятие, специализирующееся на создании программно-методических продуктов для детей.

Его адрес:

152140, г. Переславль-Залесский, ул. Полевая, 4, кв. 35.

Телефон (08535) 98400

e-mail: pervin@robot.botik.ru

Так же называется программно-методическая система, которая создана специально для курса раннего обучения информатике. Эта система включает в себя множество обучающих и игровых программ, разработанных в “Роботландии”, серию учебников для школьника и методических пособий для учителя.

Информация

Основное понятие информатики. Обозначает множество сведений, которые можно хранить, передавать и обрабатывать.

Компьютер

Электронное устройство для работы с информацией — универсальная информационная машина. Несмотря на многообразие и сложность современных компьютеров, все они имеют в своем составе:

— устройства для ввода информации (клавиатура, мышь, дисковод);

— устройства для вывода информации (дисплей, принтер, дисковод);

— устройства для хранения информации (память, диск);

— устройства для обработки информации (один или несколько процессоров).


Алгоритм

План выполнения какой-нибудь работы. Тот, кто пишет алгоритм, называется составителем, а тот, кто его выполняет, — исполнителем. Среди многообразия всех исполнителей отдельный интерес представляют различные механические и электронные устройства. Самым универсальным исполнителем является компьютер, множество алгоритмов, которые он может выполнить, неисчерпаемо.

Программа (приложение)

Запись алгоритма на языке исполнителя. Если исполнителем является компьютер, то программа — это команды процессору, записанные в файл на диске. Имя файла с программой, как правило, имеет расширение EXE или COM.

Загрузка программы

Для того чтобы программа выполнялась, ее надо переписать с диска в компьютерную память. Эту операцию выполняет операционная система, когда в ее командной строке набирается имя программы и нажимается  (клавиша выполнения).

Диск

Диск — это устройство для хранения информации при помощи магнитного слоя. Компьютеры, как правило, оснащаются винчестером — встроенным диском, способным хранить очень большой объем информации. Винчестер принято называть диском С:. Часто винчестер (или в повседневных беседах программистов — “винт”) условно делят на части, и тогда говорят не об одном, а о нескольких дисках, расположенных на одном устройстве. Чтобы отличить физический диск от условных дисков, последние называют логическими. Если винчестер содержит несколько логических дисков, то они обозначаются буквами латинского алфавита, начиная с буквы “С:”. На компьютере дедушки Фёрстова четыре логических диска — С:, D:, E: и F: (на других компьютерах число таких дисков и их обозначения могут быть другими; заметьте, что в обозначение диска, кроме буквы, входит сопровождающее ее двоеточие).

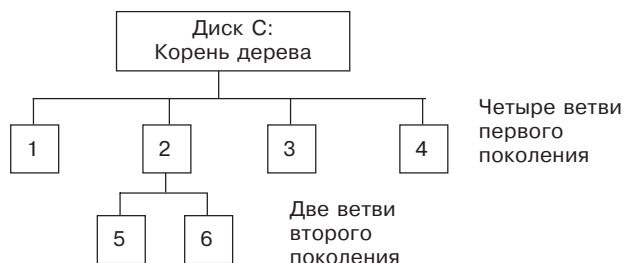
Кроме винчестера, компьютеры обычно имеют один или несколько дисководов для чтения и записи информации с гибких дисков.

Гибкие диски (или, как говорят, “дискеты”) вставляются в дисковод подобно тому, как кассета вставляется в домашний магнитофон. Объем помещаемой на них информации существенно меньше объема винчестера, но дискеты сменные, а винчестер — нет. Дисководы компьютера обозначают латинскими буквами “А:” и “В:”.

На многих современных компьютерах существуют устройства для чтения компакт-дисков. Компакт-диск (CD-ROM) — это сменное запоминающее устройство очень большого объема, и этим он очень удобен. На компакт-дисках хранят целые энциклопедии, компьютерные копии картин из замечательных музеев мира, фильмы, коллекции исторических документов и альбомы компьютерных игр. Но у компакт-дисков есть один существенный недостаток: с них можно только считывать информацию; пользователь не может писать на CD-ROM свою информацию.

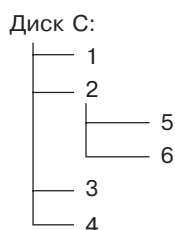
Каталог

Для того чтобы упорядочить информацию на диске, его условно разделяют на части, называемые каталогами. Это деление не совсем обычное, оно строится по принципу вхождения и подчинения. Подобно тому, как главы входят в книгу, а параграфы — в главы, одни каталоги входят в другие. Когда каталог является частью другого каталога, его называют подкаталогом. У каждого подкаталога могут быть свои подкаталоги. Если изобразить эти зависимости в виде схемы, то она будет похожа на дерево, растущее “вверх тормашками”:



На схеме изображены каталоги 1, 2, 3, 4, которые разделяют диск C: винчестера на четыре части. При этом каталог 2 имеет подкаталоги 5 и 6.

На узком листе бумаги такую схему часто рисуют более компактно:



Каталоги на дисках имеют имена, которые строятся так же, как имена файлов.

Каталоги иногда называют еще папками. Как в папках хранятся документы, так в каталогах хранятся файлы.

Файлы

Порции информации, записанные на диске, называются файлами. Файлы могут располагаться как внутри каталогов,

так и вне их. Файлы отличают друг от друга по именам. Имя файла строится из цепочки латинских букв, цифр и некоторых специальных знаков, таких, как знак подчеркивания, знак доллара, и некоторых других. Длина цепочки не может превышать 8 символов. К имени может добавляться расширение, то есть один, два или три символа с предшествующей точкой. Расширение часто используют для именования однотипных файлов. Так файлы, содержащие тексты, обычно (но не всегда!) имеют расширение TXT, файлы с программами — расширения EXE и COM. В разных каталогах могут находиться файлы с одинаковыми именами. Для поиска файла на диске в качестве его “адреса” используют запись пути к нему по дереву каталогов.

Так, например, запись

```
C:\KURSROB\KURS\micron.exe (*)
```

означает, что программа `micron.exe` расположена на диске C: в подкаталоге KURS каталога KURSROB. Иными словами, эта запись показывает, как можно разыскать программу `micron.exe`: надо найти на диске C: каталог KURSROB, внутри этого каталога — подкаталог KURS, а уж внутри этого каталога — нужную программу.

Запись, подобную (*), называют полным именем файла, или путем к файлу. Как видно из (*), полное имя состоит из имени диска, списка подчиненных каталогов и имени файла. Разделителем всех этих имен служит косая черта, а имя диска принято записывать, сопровождая его двоеточием, — C:, D: и т.д.

Текстовый файл

Файл, содержащий обычные тексты. Его называют иногда документом.

DOS

Дисковая операционная система. Специальная программа, которая упрощает работу с компьютером. DOS выступает посредником между компьютером и пользователем, берет на себя множество рутинных операций по подготовке других программ к выполнению их процессором.

В частности, для того чтобы перемещаться по дискам, каталогам и файлам, пользователь использует средства операционной системы. Поэтому можно говорить о текущем положении DOS как о пути от корня дерева файловой системы к тому конкретному файлу или каталогу, где в настоящий момент времени происходит выполнение программы или команды DOS.

Пользователь общается с DOS при помощи команд, вводимых в командную строку. Командная строка заметна на экране по символу приглашения, который DOS выводит в начало строки. Обычно приглашение имеет вид:

```
C:\> (**)
```

или

```
C:\KURSROB\KURS> (***)
```

Слева от знака “>” выводится информация о текущем положении DOS на дереве каталогов, а справа от этого знака пользователь может записывать свои команды. Запись команды завершается нажатием клавиши , после чего начинается выполнение.

Типичный пример команды DOS — имя EXE или COM файла. Такая команда вводится для запуска программы на

выполнение. При этом расширение в имени программы указывать необязательно.

Другой пример — команда `dir`. Ее используют для вывода на экран списка файлов и каталогов текущего положения DOS. Текущее положение DOS может быть изменено командой `cd` и “приклеено” к любому каталогу диска (как в примере (***)) или находится вне их (как в примере (**)).

Для управления DOS используют не только отдельные команды, но и составленные из них программы. Такие программы записывают в текстовые файлы с расширением `.BAT`. Эти файлы называются командными. Запускается командный файл так же, как и обычная программа, — по имени, записанному в командной строке DOS.

Norton Commander

Специальная программа, которая облегчает пользователю работу с компьютером. Norton Commander выводит на экран список файлов и каталогов и позволяет в отличие от DOS наглядно отображать схему их расположения на диске. Norton Commander способен выполнить практически любую команду DOS. Но, занимая своими списками большую часть экрана, эта программа оставляет внизу место и для командной строки. Поэтому, работая с Norton, можно при желании выполнять команды DOS обычным способом. Следует отметить, что Norton Commander не заменяет собой DOS, а выступает в роли посредника между DOS и пользователем.

Такую же роль, как Norton Commander, и схожую структуру имеют некоторые другие часто используемые программы, работающие под DOS, — Volkov Commander, Farmanager (или просто Far) и др.

Windows

Многооконная операционная система. Хотя она оперирует теми же объектами, что и DOS — файлы и папки (так называются каталоги в Windows), — структура и внешнее представление информационных объектов в этой операционной системе иное. Основные черты Windows: пиктографическое меню информационных объектов, существенно возросшая роль мыши как периферийного устройства, реальная многозадачность — возможность одновременной работы с несколькими приложениями.

Вместе с тем, работая с теми же объектами, что и DOS, эта операционная система требует от пользователя владения теми же основными операциями обработки информационных объектов — копирование и перемещение файлов, открытие и закрытие документов и папок, запуск программ.

Обе программные системы — DOS и Windows — называют программными оболочками.

Приложения, описываемые здесь, в одинаковой мере работают и в той, и в другой программных оболочках. С точки зрения основной задачи этого курса — обучение принципам и приемам конструирования приложений — различия двух программных оболочек значения не имеют.

Интерфейс

Способ организации диалога между программой и пользователем.

Пользователь знаком с интерфейсом программы, если знает, что нажимать на клавиатуре или на каких областях экрана надо щелкать мышкой, для того чтобы вызвать у программы нужные действия.

Несмотря на то, что интерфейс как способ взаимодействия с программой постоянно пытаются стандартизировать, очень часто одни и те же действия в разных программах выполняются по-разному.

Простейшими элементами интерфейса являются:

- курсор;
- меню выбора;
- экранные кнопки;
- пиктограммы;
- окна;
- “горячие клавиши”.

Курсор

Метка на экране компьютера, перемещаемая при помощи мыши или стрелок клавиатуры. Курсор играет роль электронной указки и может иметь разную форму.

Меню выбора

Список объектов, задающий различные действия программы.

Пользуясь стрелками клавиатуры или мышью, пользователь может отметить курсором нужный объект. Программа начинает работать в соответствии со сделанным выбором после дополнительного нажатия на клавишу **Enter** (клавишу выполнения) или сразу в момент выбора.

Экранные кнопки

Кнопки, нарисованные на экране компьютера. При работе с мышью экранные кнопки дополняют (или заменяют) клавиши на клавиатуре. Для того чтобы “нажать” экранную кнопку, нужно установить на нее курсор мыши и коротко нажать (“щелкнуть”) левую кнопку мыши. Вместо левой кнопки мыши в программах Роботландии можно использовать клавишу **Enter** на клавиатуре.

Пиктограмма

Условное обозначение объекта. Широко используется как в информатике, так и в повседневной жизни. Общеизвестный пример — знаки дорожного движения.

Окна

Работа с программой может выполняться либо непосредственно на экране, либо во всплывающем окне. Окна бывают информационными и рабочими.

“Горячие клавиши”

Список управляющих клавиш, наиболее часто используемых при работе с программой.

Почти во всех программах Роботландии:

аккорд **Ctrl** + **Break** — служит для выхода из программы,

аккорд **Ctrl** + **BackSpace** — для выполнения откатки,

аккорд **Ctrl** + **Enter** — для выполнения накатки,

F1 — для вызова справки.

При работе с мышью всегда имеют место следующие соответствия:

левая кнопка мыши — ,

правая кнопка мыши — .

Рабочее поле и пиктограммы

Программы, используемые в курсе “Компьютерное конструирование”, работают как в двухпанельной программной оболочке DOS, так и в многооконной среде Windows. Ничто не мешает учителю в нужный момент вызвать Norton Commander (или Far, или другую имеющуюся на машине оболочку DOS), чтобы поработать в среде DOS (если пользователь к ней привык). Однако детям целесообразно сразу представить рабочее поле более современной оконной среды Windows: большинство школьных (и тем более домашних) компьютеров располагают такими средствами.

Готовя компьютер к занятию с младшими школьниками, полезно позаботиться о презентабельности рабочего стола. Зачастую пользователь, злоупотребляя возможностями Windows, размещает на рабочем столе большое число ярлыков. При индивидуальном использовании компьютера насыщенность экрана не существенна и может определяться привычкой или персональными предпочтениями. Новичкам-малышам лучше предложить обстановку, в которой для рассеяния их внимания будет меньше предпосылок. Целесообразно убрать (возможно, на время) те пиктограммы, которые на занятиях не понадобятся.

С другой стороны, рекомендуется вынести на рабочее поле пиктограмму `konstr.ico`, которая поставляется вместе с программным пакетом “Конструктор”.

Для пользователя, умеющего манипулировать ярлыками рабочего стола, этой информации, вообще говоря, достаточно. Тем, кто будет делать установку пиктограммы впервые, рекомендуется воспользоваться приводимой инструкцией, которая находится в файле `read.me`, хранящемся в корне пакета “Конструктор”.

Имеет смысл выбрать на время занятий подходящие экранные обои — спокойную, не раздражающую картинку. Лучше вообще иметь чистый фон, чем дополнительный отвлекающий фактор.

Войдя в “Панель управления”, выполните необходимые настройки, чтобы установить крупные значки (пиктограммы-ярлыки) и крупные символы. Это облегчит детям ориентацию в рабочем поле.

Завершив системные установки, надо опробовать запуск пакета “Конструктор”, а затем пройти по страницам этого пакета. О навигации по пакету можно прочитать в документации к программе “Монитор” (запуск — щелчок по пиктограмме “Описание Монитора” на заглавной, конфигурационной странице пакета). Для подготовки к первым занятиям достаточно ознакомиться пока с “навигационным”, четвертым разделом описания.

Кроме того, исчерпывающую информацию по “Монитору” дает первая глава учебника, которая целиком посвящена “Монитору”.

ГЛАВА 1. МОНИТОР

1. Знакомьтесь — Монитор!

Дедушка Фёрстов, хотя и знает неутомный характер своих внуков Тима и Дины, предпочитает, однако, чтобы в его кабинет они не врвались, а входили, не торопясь и не споря о том, кто первым сядет за клавиатуру. Впрочем, и ребята тоже хорошо знают своего дедушку. Поэтому они так осторожно зашли в комнату, что дедушка, увлекшийся просмотром своей статьи на компьютерном экране, даже не заметил их появления. Диночка тихонько кашлянула и сказала:

— Дедушка, мы пришли. Как и договаривались. Ведь сегодня ты хотел рассказать нам про программу *Монитор*.

— Что ж, заходите, внуки мои. Рассаживайтесь, — сказал дедушка Фёрстов, приглашая детей к столу с компьютером.

И, пока они усаживались по обе стороны от дедушкиного кресла, он успел привести в порядок компьютерный экран.

— Да, действительно. У меня есть предложение: серьезно исследовать роботландский *Монитор*, ведь мы им пользуемся почти каждый день.

Тим выразил общее согласие:

— Да, в нем очень удобно работать. Если я хочу запустить какую-нибудь программу, я просто щелкаю мышкой по нужной кнопке. Когда у нас не было *Монитора*, приходилось долго искать программы на диске по именам файлов, записанным английскими буквами.

Дедушка Фёрстов был рад, что так быстро встретил понимание:

— Название этой программы происходит от французского слова *moniteur*, то есть “вожатый”. Или, еще точнее, “проводник”, тот, кто ведет вас по новым, неизве-

данным дорогам. Такие же задачи выполняет и программа *Монитор*: она позволяет уверенно путешествовать от программы к программе по меткам, которые устанавливаются на пути, словно придорожные столбы с метками-указателями.

Экран этой программы похож на рабочий стол, где все необходимые инструменты всегда под рукой. И, главное, если чего-то не хватает, всегда положить на него новый инструмент, другими словами, добавить новую кнопку.

Дина, боясь, что дедушка сразу предложит высокий темп, предупредила его:

— Я пыталась это делать, начала читать программную документацию, но, честно говоря, не все поняла, и у меня не получилось.

Дедушка успокоил:

— Надеюсь, что сегодня все встанет на свое место, если двигаться, не торопясь. *Монитор* имеет много интересных возможностей, и я думаю, что нам неплохо бы с ними познакомиться поближе.

Тим был наготове:

— Можно, я? Я знаю, как загрузить на экран *Монитор* с роботландским пакетом *Конструктор*.

Тим сказал так потому, что увидел на экране пиктограмму с названием *Конструктор*. Он щелкнул по ней, и на экране появилось знакомое ребятам поле с кнопками. В заголовке страницы выделялось слово “Конструктор”. Но, поскольку дедушка уже говорил ребятам о пакете *Конструктор*, с которым им придется работать целый учебный год, оно их тоже не удивило:



Дедушка предложил ребятам для начала рассказать о *Мониторе* то, что они уже знают про эту программу, и попросил Дину сказать первые слова. Не смущаясь, внучка заговорила:

— У *Монитора* верхняя полоска своя собственная, “мониторская”, а остальное поле содержит кнопки с рисунками и надписями — это, как ты сам говорил, наш рабочий стол.

— А что же находится в верхней полоске *Монитора*?

В разговор включился Тим, молчать было трудно:

— Пять кнопок, название пакета и часы, они “тикают” справа. Первая кнопка с надписью “STOP” понятна. Если щелкнуть по ней мышкой, то *Монитор* заканчивает свою работу. Дальше...

Он щелкнул по следующей кнопке и, прочитав выпавший в окошечко текст, сказал:

— Вторая кнопка с нарисованным мальчиком вызывает на экран текст, в котором рассказано об авторах — о тех, кто написал программу *Монитор*, а третья кнопка, с английской буквой “i”, — это, должно быть, “информация”. О чем?

Новый щелчок, и Тим с Диной с удивлением увидели свою фамилию.

Дедушке пришлось пояснить:

— Эта кнопка выводит на экран текст, в котором говорится, что пакет *Конструктор* куплен Фёрстовым и имеет регистрационный номер 24012602003. Таким образом авторы пакета защищают свои права. Если бы в этом тексте говорилось не о Фёрстове, а, например, о Кукушкине, это означало бы, что мы приобрели пакет незаконно.

Дедушка Фёрстов не стал развивать тему незаконного распространения программ, а продолжил обзор кнопок в верхней строке *Монитора*.

— Четвертая кнопка с нарисованным внутри нее карандашом вызывает *Редактор пиктограмм*. Ты ведь знаешь, Дина, что такое пиктограмма?

— Конечно. Пиктограмма — это условное обозначение.

— Можно так сказать. В *Мониторе* пиктограммы — это рисунки на кнопках. Если на кнопке изображен карандаш, то она, как можно догадаться, запускает “рисовальную” программу.

Тим обоснованно предположил:

— Значит, используя *Редактор пиктограмм*, мы можем изменить внешний вид кнопок, расположенных на экране?

— Верно. И этим мы будем, конечно, заниматься. Наконец, последняя кнопка в верхней полоске *Монитора* — самая важная: она позволяет добавлять на рабочий стол новые кнопки, удалять или изменять существующие, менять окраску экрана. Будем называть эту кнопку *Операции*.

Дина попыталась найти объяснение этому названию:

— На этой кнопке изображен красный крест, как на машине “скорой помощи”. А врачи-хирурги часто делают большим операции. Поэтому ты и предлагаешь, дедушка, такое название?

— Пожалуй, сравнение с врачами здесь не очень удачно. Лучше уж сравнивать это название с набором тех действий, которые выполняют токари или кузнецы, изготавливая новое изделие. Вся их работа состоит из отдельных технологических операций. Точно так же и нам, изготавливая новые или “ремонтируя” готовые пиктограммы, придется выполнять операции, предусмотренные *Редактором пиктограмм*. Сегодня мы их рассмотрим повнимательнее.

А пока подведем итог самого первого обсуждения. Изображение *Монитора* на экране состоит из двух частей: служебной (верхней) полоски и рабочего стола пользователя. В служебной полоске располагаются название пакета, часы и служебные кнопки. На рабочем столе находятся кнопки пользователя. Они предназначены для запуска разных программ, или, как говорят, приложений. В отличие от служебных кнопок пользовательские можно редактировать, то есть:

— менять положение на экране,

— менять внешний вид,

— менять запускаемое этой кнопкой приложение.

Кроме того, кнопку пользователя можно вообще удалить с экрана и, наоборот, на рабочий стол можно добавить новые кнопки.

Дина поинтересовалась:

— А сколько же кнопок можно поместить на рабочем столе?

— Не больше двенадцати.

Ой, так мало? Значит, из *Монитора* можно запускать только 12 программ?

Дедушка назвал этот вопрос Дины очень важным и начал объяснять:

— Нет, такого ограничения в *Мониторе* не существует. Дело в том, что одни кнопки могут запускать приложения, а другие — открывать новые экраны — страницы с новыми кнопками. Таким образом, *Монитор* может запускать сколько угодно программ, или по крайней мере достаточно много.

Тут задал свой вопрос Тим:

— А как догадаться, какая кнопка запускает приложение, а какая выполняет переход на новую страницу?

— Вот посмотри на экран. На рабочем столе сейчас расположены пять кнопок. Все они служат для перехода к новым страницам *Монитора*.

— А как ты это определяешь?

Дедушка ответил, показывая на экран:

— Во-первых, кнопка-страница “подмигивает”, а во-вторых, текст под ней заканчивается тремя точками. Это принятое в *Мониторе* обозначение для страницы.

2. Работа в Мониторе

И Тиму, и Дине подведенные дедушкой итоги были ясны. Все же Тим попытался уточнить:

— Значит, если я щелкну мышкой по кнопке “Мудрый Крот...”, эта программа не запустится?

— Как мы договорились, многоточие за названием кнопки означает, что для *Мудрого Крота* в *Мониторе* отведена целая страница и щелчок по кнопке заставит *Монитор* перейти на нее.

— Попробую. Перевожу стрелку-курсор на пиктограмму *Мудрого Крота* и нажимаю левую кнопку мыши. Готово! Смотрите, что у меня получилось:



И, объясняя самому себе (и Дине) эту картинку, Тим сказал:

— На странице *Мудрого Крота* расположены две кнопки. И что интересно: ни на одной из этих кнопок нет многоточий. Значит, обе они — не обращения к страницам, а вызовы программ.

— Верно, — ответил дедушка Фёрстов. — Но, если говорить точнее, первая из них вызывает саму программу, а вторая выводит на экран ее описание (правда, вывод описания тоже осуществляется, как вы увидите, вызовом специальной программы).

Обратите внимание на изменения, которые произошли в служебной полоске *Монитора*, когда Тим щелкнул по кнопке с пиктограммой “кротовой” страницы.

Дина отметила:

— Первая кнопка “STOP” заменилась на кнопку с изображением отогнутого листка.

Дедушка похвалил внимательность внучки:

Это означает, что нажатие на нее вернет *Монитор* назад на предыдущую страницу. *Монитор*, таким образом, сделал закладку. Кстати, когда вы читаете книгу, вы тоже часто пользуетесь закладками.

— Да, пользуемся, — ответила Дина. — Только страницы при этом не загибаем. Так с книгами обращаться нельзя.

— Совершенно справедливо, Дина. Такое можно позволить только в изображении пиктограмм, но никак не в настоящей книге.

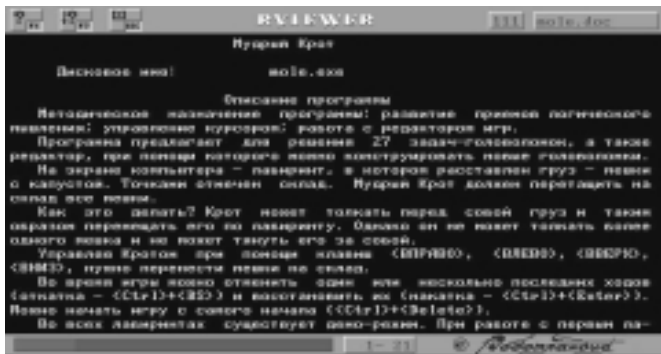
А Тим нашел еще одно изменение внешнего вида страницы:

— Еще название пакета *Конструктор* заменилось на текст “Мудрый Крот”.

— И это верно. Теперь это уже не заголовок всего пакета программ, а название страницы, на которую нас перевел *Монитор*. Оно берется из названия кнопки, которую мы нажали на первой странице. Три точки при этом отбрасываются.

Тим предложил:

— Сначала посмотрим описание *Крота*. Нажимаю кнопку с этой надписью...



Пока дети рассматривали экран с описанием *Мудрого Крота*, дедушка отметил:

— *Монитор* запустил программу просмотра и показывает текстовый файл (или документ) `mo1e.doc`.

Дина удивленно спросила дедушку:

— А как ты узнал, что этот файл так называется?

— Посмотри: его имя записано справа в верхней служебной полоске программы просмотра. Еще в этой полоске обозначены “горячие клавиши”:

F1 — для вызова списка управляющих клавиш,

F7 — для поиска в файле нужного текста,

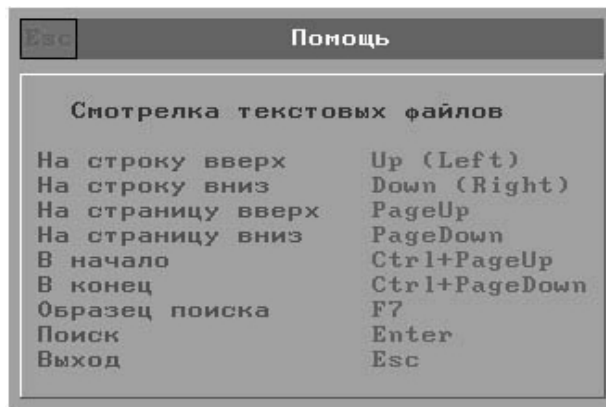
Esc — для выхода из программы просмотра.

— Почему-то не видно курсора мыши! — на этот раз удивился Тим.

— Эта программа управляется только с клавиатуры.

Тим не стал возражать против этого ограничения:

— Ну тогда нажимаю клавишу **F1**, чтобы узнать, как это делается:



Дедушка решил дать пояснения к этой экранной таблице:

— Нажимая стрелки **↑**, **↓**, можно прокручивать текст на экране. Для быстрых перемещений по тексту можно использовать клавиши



Заметьте, что, когда вы перемещаете окно просмотра, его примерное положение в тесте отмечает прямоугольник в нижней служебной полоске. Точное положение там же показывают числа, записанные через черточку. Запись “1—21”, например, означает, что на экране видны строки с первой по двадцать первую. А рядом с именем просматриваемого файла указано общее число строк в нем.

Дина оценила программу—“смотрелку”:

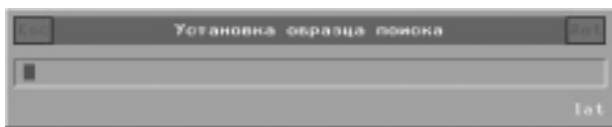
— Управление очень простое. Все понятно, кроме, пожалуй, двух строк в списке управляющих клавиш:

Образец поиска **F7**

Поиск **Enter**

— Эти операции связаны с поиском в файле небольших порций текста. Первая из них задает образец поиска, а вторая заставляет программу показывать места его вхождения в файл. Давайте попробуем что-нибудь поискать. Нука, Тим, нажми клавишу **F7**.

— Готово! На экране появилось окошко:



— Теперь ты можешь записать в него то, что хочешь найти. Каждое последующее нажатие на **Enter** будет отмечать в тексте новое вхождение записанного образца, если, конечно, он там присутствует.

Тим написал в полоске ввода слово “Крот” и нажал несколько раз на **Enter**. Результатами своих действий он был явно доволен.

— Теперь я нажимаю на клавишу **Esc** и возвращаюсь в *Монитор* из программы просмотра. Подвожу курсор мыши к кнопке *Мудрый Крот*, щелкаю мышкой, и вот на экране моя любимая игрушка!

Но дедушка умирал его пыл:

— Предлагаю сегодня оставить *Крота* в покое и продолжить изучение *Монитора*.

Тим признался:

— Действительно, я чуть увлекся, а по *Монитору* у меня еще куча вопросов.

Прежде чем выяснить вопросы внука, дедушка сказал:

— Вы знаете, что я всегда внимательно читаю описание программ, перед тем как начать с ними работать. Работу с *Монитором* можно ускорить, если, кроме мышки, использовать еще и клавиатуру. Например, для выхода из *Монитора* или возврата на предыдущую страницу можно наряду с первой служебной кнопкой использовать любой клавиатурный аккорд из приведенных ниже:

Ctrl + **X** **Ctrl** + **Q**

Ctrl + **S** **Ctrl** + **Break**

Другим служебным кнопкам соответствуют следующие клавиатурные нажатия:

Кнопка “Авторы пакета”

— **Ctrl** + **R** — **Ctrl** + **P** (“P” — русская)

Кнопка “Владелец пакета”

— **Ctrl** + **I** — **Ctrl** + **И**

Кнопка “Редактор Пиктограмм”

— **Ctrl** + **P** — **Ctrl** + **П**

Кнопка “Операции”

— **Ctrl** + **O** — **Ctrl** + **О** (“O” — русская)

Стоит добавить, что почти во всех программах Роботландии клавиша **Enter** соответствует левой кнопке мыши, а

клавиша **Esc** — правой. Первую клавишу роботландцы часто называют клавишей выполнения, а вторую — клавишей перехода.

Тим глубокомысленно заметил:

— Ну, я думаю, это дело привычки: кому-то больше нравится работать с мышью, кому-то с клавиатурой.

— Тем не менее замечен интересный факт: чем больше пользователь работает с программой, тем чаще он использует клавиатуру — с ней работать быстрее.

3. Анатомия Монитора

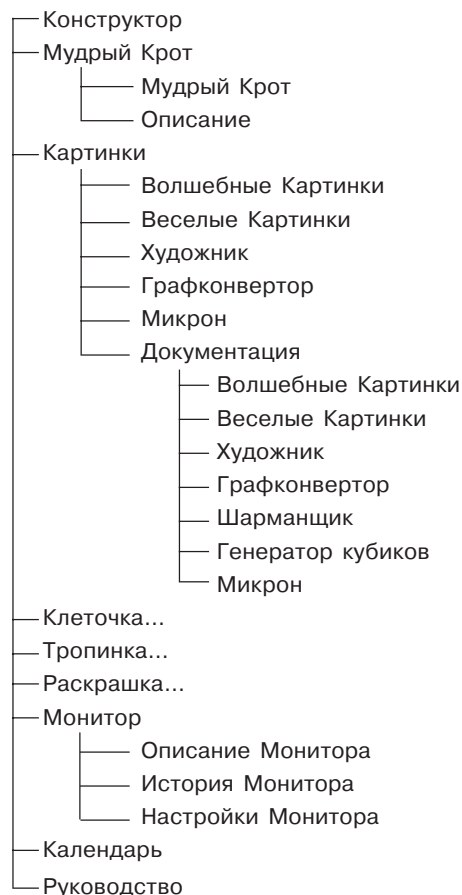
Дедушка взял с края стола заготовленную страничку с набросанными на ней линиями и надписями:

— Устройство страниц *Монитора* я изобразил такой схемой (см. следующую страницу).

В схеме я использовал линии для указания принадлежности одних объектов другим. По ней легко определяется, например, что кнопка *Микрон* находится на странице *Картинки...*, вход в которую, в свою очередь, выполняется со страницы *Конструктор*. В то же время по схеме легко понять, что кнопки *Художник* и *Шарманщик* лежат на одной странице *Картинки*.

Страница “Монитор” содержит, как легко видеть, три документа:

- инструкцию по *Монитору*,
- историю изменений *Монитора* от версии к версии,
- настройки *Монитора* (по этой пиктограмме вызывается текстовый редактор *Микрон* для изменения файла с настройками *Монитора*).



Дина еще раз продемонстрировала, насколько она внимательна:

— Страницы *Мудрый Крот...* и *Картинки...* на этой схеме описаны полностью, а страницы *Клеточка...*, *Тропинка...* и *Раскрашка...* на твоей схеме, дедушка, почему-то не “раскрыты”.

Тут дедушка хитро улыбнулся:

— Я оставил эту работу для вас...

Задача 1. Пользуясь образцом схемы, предложенной дедушкой Фёрстовым, нарисовать полную схему пакета *Конструктор*.

Для выполнения задания придется вызвать на экран пакет и “походить” по его страницам.

Пока Дина и Тим дружно трудились над поставленной задачей, дедушка тоже не терял времени. И, как только ребята завершили свое задание и обсудили результаты с дедушкой, он показал им новую, более подробную схему того же *Монитора*:

```
C:\
├── KURSROB — каталог пакета Конструктор
│   ├── go.bat — файл запуска Монитора
│   ├── desk.exe
│   └── maindesk.exe ] Монитор
│   ├── desk.ini — файл настроек Монитора
│   ├── ini.bat — файл запуска редактора настроек
│   ├── dp.exe
│   └── dp.pic ] Редактор Пиктограмм
│   ├── Rview.exe — “смотрелка” текстовых файлов
│   ├── adp.exe ] вспомогательные программы
│   └── rdd.exe ] (утилиты Монитора)
│   ├── read.me — описание установки и запуска
│   └── DOC — Каталог инструкций по Монитору
│       ├── book.doc — неформальное описание Монитора
│       ├── desk.doc — описание Монитора
│       ├── change.doc — история развития Монитора
│       └── dp.doc — описание Редактора Пиктограмм
│   └── ICON — Каталог пиктограмм Монитора
│       ├── doc.cut — пиктограмма “документ”
│       ├── mole.cut — пиктограмма для Мудрого Крота
│       ├── pic1.cut — пиктограмма для Волшебных Картинок
│       ├── pic2.cut — пиктограмма для Волшебных Картинок
│       └── gen.pic — пиктограмма для Генератора Кубиков
│       ...
│   └── SYS — Вспомогательный каталог Монитора
│       ├── deskp.000
│       ├── deskp.001
│       └── deskp.002 ] описание страниц Монитора
│       ...
│       ├── deskt.cut
│       ├── desk.dsk
│       └── mem.go ] вспомогательные файлы Монитора
│   └── BLOKNOT
│       ├── note.abc — база данных алфавитного блокнота
│       ├── note.col — база данных календарных записей
│       ├── note.doc — описание программы
│       ├── note.exe — запуск программы
│       ├── note.ini — файл настроек Блокнота
│       ├── note.lst — заголовок БД календарных записей
│       └── note.pcx — файл с картинкой
│   └── KURS
│       ├── micron.exe — редактор Микрон
│       ├── Micron.doc — описание Микрона
│       ├── paint.exe — редактор Художник
│       ├── paint.doc — описание Художника
│       ├── sharm.exe — редактор Шарманщик
│       ├── sharm.doc — описание Шарманщика
│       ├── GS — Каталог Графконвертора
│       ├── 1 — Каталог Конструкторов
│       ├── ABC — Каталог Волшебных Картинок
│       ├── KL — Каталог Клеточки
│       └── KP — Каталог Раскраски
```

```
├── PATH — Каталог Тропинки
├── MOLE — Каталог Мудрого Крота
│   ├── mole.exe — Мудрый Крот
│   └── DOC — Каталог документации
│       └── mole.doc — описание Мудрого Крота
├── МЕТОДИКА
│   ├── clnd.txt — календарный план курса
│   ├── part1.txt — Первая глава (занятие 1)
│   ├── ...
│   ├── part6.txt — Шестая глава (занятие 6)
│   ├── teach.txt — Учительский текст
│   └── work.bat — файл запуска
└── MIM
    ├── mim.dir — оглавление
    ├── mim.exe — программа МикроМира
    ├── mim.hlp — справочник МикроМира
    ├── read.me — описание МикроМира
    ├── reklam.txt — рекламный текст
    ├── tune.exe — настройщик МикроМира
    └── work.bat — файл запуска
```

— Здесь показано, как устроен пакет *Конструктор* на нашем винчестерском диске.

Тим бесспорно признал:

— Да, эта твоя схема гораздо сложнее. Но тут обязательно нужны пояснения.

Дедушка был готов к этому:

— Эта схема показывает, как на самом деле располагаются файлы, связанные с пакетом *Конструктор*, на жестком диске C: . Весь пакет находится в каталоге KURSROB. Он состоит из двух частей.


Первая, вспомогательная, связана с работой *Монитора*, а вторая — с файлами отдельных конструкторов. Для работы *Монитора* необходимы каталог ICON (в нем располагаются пиктограммы используемых кнопок) и каталог SYS (в нем расположены файлы с описанием страниц Монитора и некоторые другие вспомогательные файлы).

Вторая часть находится в каталоге KURS. Этот каталог содержит программы общего пользования (их используют сразу несколько конструкторов) и каталоги с самими конструкторами. Я подробно изобразил только каталог, содержащий *Мудрого Крота*.

Пока мы можем отложить эту схему в сторонку, но мы еще к ней вернемся, когда будем учиться строить свои собственные пакеты на основе *Монитора*. А сейчас давайте займемся редактированием.

4. Редактирование Монитора

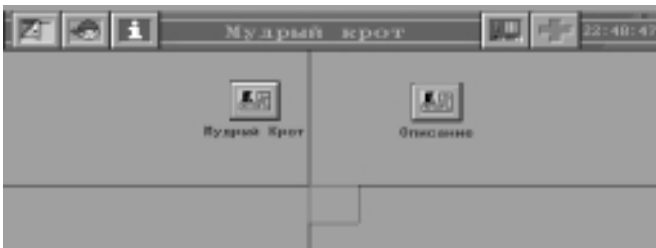
4.1. Перемещение кнопок

— Вот простая операция: посмотрим, как изменить положение кнопки на рабочем столе *Монитора*. Делается это очень просто. Нужно установить курсор мыши на перемещаемую экранную кнопку и нажать не левую, а правую кнопку мыши, или, что то же самое, нажать клавишу  на клавиатуре.

За время рисования схемы Дина и Тим двигали свои стулья. Теперь ближе к коврику с мышкой оказалась Дина. Поэтому она взялась за дело:

— Сейчас я попробую это сделать на странице *Мудрого Крота*. Готово! Кнопка осталась на месте, а курсор мыши

заменился на прямоугольник размером с кнопку, да еще с перекрещенными линиями:



Дедушка продолжил указания:

— Теперь, двигая мышь, ты можешь выбрать новое место для кнопки, а перекрестье тебе поможет определить его более точно. Впрочем, для небольших точных перемещений лучше использовать не мышь, а стрелки клавиатуры.

Давайте установим сейчас кнопку с описанием точно под кнопкой вызова программы *Мудрого Крота*.

Это действие Дина выполнила без труда:

— Готово! Перекрестье и клавиатура, в самом деле, здорово помогают.

Тим остался недоволен:

— Но кнопка-то все равно расположена на старом месте!

Дедушка помог:

— Чтобы она встала на новое место, теперь нужно нажать левую кнопку мыши. Или клавишу выполнения.

Дина сделала и это:

— Нажимаю **Enter**. Получилось!

Дедушка продолжал:

— В *Мониторе* предусмотрено автоматическое перемещение кнопок в “стандартные” места. Оно выполняется при помощи клавиши **F2** на клавиатуре. Если сейчас ее нажать, то кнопка “Описание” вернется на прежнее место, потому что изначально она располагалась именно в “стандартном” месте рабочего стола.

Дина попыталась уточнить:

— А где на экране находятся места, которые ты называешь стандартными?

— Таких мест на экране 12, по наибольшему числу кнопок, разрешаемых на странице. Они невидимы и располагаются симметрично на рабочем столе *Монитора*. Если пользователь принялся искать новое место для кнопки при помощи прямоугольника с перекрестьем, а потом раздумал, он всегда может отказаться от режима перемещения, нажав еще раз правую кнопку на мыши или клавишу перехода на клавиатуре.

4.2. Копирование кнопок

Дина размечталась:

— Эх, хорошо бы перенести кнопку на другую страницу *Монитора*! Помните, как, работая с программой *Блокнот*, мы научились переносить записи с одной даты в календаре на другую?

Дедушка поддержал эту замечательную идею:

— Что ж, это можно делать. Причем средствами, очень похожими на те, о которых вспомнила Дина, — и добавил: — Только не в режиме перемещения, а в режиме копирования.

Тим спросил:

— И как же это делается?

— В *Мониторе* есть область памяти, своеобразный “карман”, для временного хранения файла с пиктограммой одной кнопки. Эту специальную память называют буфером. Буфер никак не отображается на экране, он встроен в программу управления *Монитором*. (“Точно, как в *Блокноте*”, — подумала про себя Дина.)

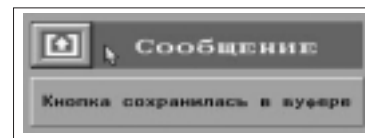
Для того чтобы запомнить кнопку в буфере, нужно нажать клавишу **+** на клавиатуре в то время, когда курсор мыши расположен на кнопке. А для того чтобы переписать кнопку из буфера на рабочий стол, нужно нажать на клавиатуре клавишу **-**, и кнопка появится на экране в том месте, на которое указывает стрелка курсора.

Операцию переноса кнопки с места на место с помощью буфера можно делать и внутри одной страницы, но, конечно, более полезна такая операция для случая, который отметил Дина, — перенос кнопки на другую страницу.

Дина, подбодренная дедушкиной поддержкой, поспешила действовать:

— Попробую-ка я переместить кнопку *Описание* на предыдущую страницу. Ставлю курсор на кнопку, нажимаю клавишу **+**.

Появилось окошко с сообщением:



Дедушка пояснил:

— Все нормально. Это просто информация о том, что операция прошла успешно.

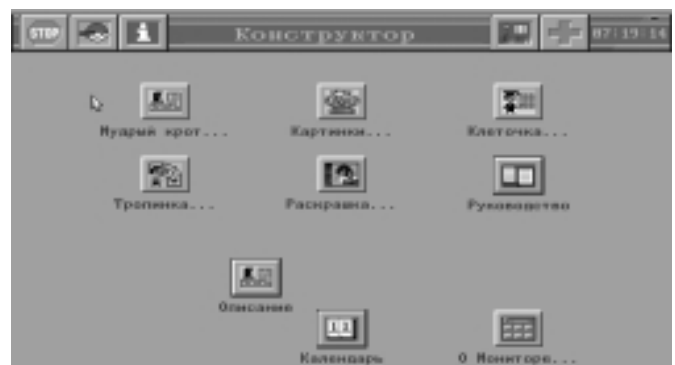
— А как теперь можно убрать это окошко с экрана? — поинтересовался Тим.

— Одним из трех способов:

- нажать экранную кнопку в левом верхнем углу окна с сообщением;
- нажать правую кнопку мыши в любом месте экрана;
- нажать клавишу перехода на клавиатуре.

Дина отреагировала на это указание:

— Хорошо. Нажму правую кнопку мыши. Окно пропало, теперь перехожу на предыдущую страницу *Монитора*, подвожу курсор к свободному месту, нажимаю клавишу **-**. Смотрите, кнопка действительно появилась на экране!



— А старая кнопка на странице *Крота* пропала? — спросил Тим.

— Нет, но ты можешь удалить ее.

— Нет, думаю, что удалять ее оттуда не стоит: там как раз ей самое место. Нужно, наоборот, удалить кнопку на первой странице. А, кстати, как удаляются кнопки?

4.3. Удаление кнопок

— Для удаления придется воспользоваться служебной кнопкой *Монитора* с красным крестом на ней — *Операции*.

Тим вспомнил, какие предположения делала сестра по поводу названия этой кнопки, и шуточно сказал:

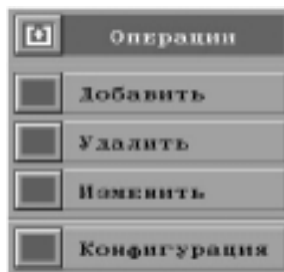
— Ага, придется делать операцию, как в больнице!

Дина оценила доброе настроение брата и предложила ему поменяться местами у клавиатуры:

— Садись. Теперь твоя очередь.

Тот, разумеется, не возражал:

— Нажимаю на красный крест, и вот что у меня получилось:



Это, как я догадываюсь, и есть меню операций. Сейчас нам нужен раздел *Удалить*. Щелкая мышкой по этому разделу. Меню с экрана пропало, а мышьяная стрелка превратилась в корзинку, вроде той, что стоит в этом кабинете для ненужных бумаг.

Дедушка включился в этот поток высказываний внука:

— Теперь нужно установить корзину на лишнюю кнопку и нажать левую кнопку мыши.

— Я удалил кнопку, но курсор по-прежнему имеет форму корзинки.

— Это значит, что можно продолжать удалять другие кнопки.

— Но мне ничего больше удалять не надо!

— Тогда нажми правую кнопку мыши, и курсор вернет свою привычную форму.

— Значит, можно нажать и клавишу перехода, ведь это то же самое! Да, курсор снова стал стрелкой.

Дедушка предложил еще одну небольшую серию комментариев:

— По поводу клавиш и мышки хочу заметить, что выбор раздела в меню операций можно выполнять и с клавиатуры:

Добавить — **Ctrl** + **Д**

Удалить — **Ctrl** + **У**

Изменить — **Ctrl** + **И**

Конфигурация — **Ctrl** + **К**

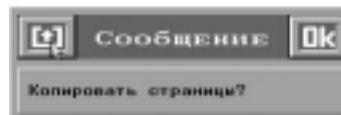
И еще одно замечание. Копирование и удаление кнопок, изображающих переход на новые страницы, имеет свою особенность. Ведь за такой кнопкой скрывается гораздо больше информации, чем за обычной. Вот, например, Тим, скопируй кнопку *Картинки...* с первой страницы *Монитора* на страницу *Мудрого Крота*.

— А я ничего не испорчу?

— Не волнуйся, вперед! Я же с тобой!

Тим начал осторожно:

— Устанавливаю курсор на кнопку *Картинки...* Нажимаю клавишу **+**... Пока все, как обычно. На экран вывелось сообщение: “Кнопка сохранилась в буфере”. Теперь перехожу на страницу *Мудрого Крота* и нажимаю клавишу **-**. Появилось окно с новым сообщением. И не с одной, а с двумя кнопками:



Дедушка решил подбодрить Тима:

— Две кнопки нужны потому, что тебе задан вопрос и ожидается один из двух возможных ответов. Левая кнопка служит для ввода ответа “нет”, правая, с надписью “Ok”, — для ввода ответа “да”.

— А что мне выбрать?

— Давай попробуем оба варианта и пойдем, в чем разница. Сначала ответ “да”.

Тим так и сделал:

— На экране появилась копия кнопки *Картинки...*

— Теперь поставь рядом такую же кнопку, но на вопрос “Копировать страницы?” ответь “Нет”.

Тим попытался уточнить:

— Мне надо перейти на первую страницу и снова выполнить копирование?

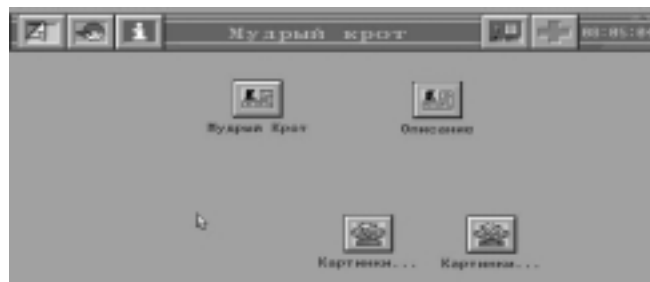
Дедушка ответил:

— Этого делать не надо. Ведь нужная кнопка уже хранится в буфере. Она исчезнет оттуда лишь тогда, когда на ее место ты запишешь другую.

— Это хорошо, — признал Тим. — Итак, теперь у меня на странице *Крота* две одинаковых кнопки.

Дедушка в связи с этим сказал:

— Верно, но первая получилась с копированием страниц, а вторая — нет.



Каждая из твоих новых кнопок задает переход на другие страницы. Перейди сначала на страницу, задаваемую второй кнопкой.

— Хорошо, перешел. Эта страница пуста.

— А теперь перейди на страницу, определяемую первой кнопкой.

Тим выполнил и это действие:

— Перешел. На ней много других кнопок... Эта страница выглядит точно так же, как если бы мы нажали кнопку “Картинки...” на первой странице, а не на странице *Мудрого Крота*.

— Теперь ты понял смысл вопроса на экране?

— Да. Если я запрещаю копировать страницы, то новая кнопка определит пустую страницу, а если разрешу, то получится полная копия.

Довольный полным и содержательным ответом Тима, дедушка задал вопрос:

— Смог бы ты нарисовать теперь измененную схему страницы *Мудрого Крота*?

— Попробую... — сказал Тим и подвинул к себе страничку, на которой раньше дедушка изобразил схему пакета *Конструктор*.



Рисунок был сделан при непосредственном участии Дины. Когда этот совместный труд был закончен, Тим сказал:

— Последняя закорючка в схеме означает, что на странице ничего нет: она пуста. У меня есть вопрос: когда я копирую кнопки, копируются ли программы на диске тоже?

Дедушка понял, что от него требуется ответ, более развернутый, чем короткое отрицание:

— Конечно, нет. Копируется лишь информация об их расположении на диске. Информация, которую называют ссылкой. Но тем не менее ты можешь запускать программы откуда угодно, используя как оригиналы кнопок, так и их копии. Ну а теперь удали обе эти кнопки со страницы *Крота*, ведь они там совсем не нужны.

Тим выполнил просьбу дедушки, и ребята сделали такое наблюдение, по поводу которого Дина, внимательно следившая за всеми действиями брата, сказала:

— Вторая кнопка удалась без проблем, а при удалении первой возник вопрос:



Дедушка объяснил:

— *Монитор* предупреждает вас об ответственности. Ведь на самом деле будет удалена не только одна кнопка страницы, но и все другие, расположенные на удаляемой странице. Иными словами, удалится информация, которую ты сейчас не видишь.

Но у Тима осталась еще одна неясность:

— Вторая кнопка тоже вызывает страницу, но вопроса не было.

— Страница, которую вызывает вторая кнопка, пуста, поэтому никаких скрытых потерь при удалении не произойдет.

Тим был удовлетворен.

— Спасибо, дедушка. Все понятно. Надеюсь, что наши опыты не повредили оригинал *Картинок* на первой странице?

— Конечно, нет, ведь мы работали с копией.

4.4. Создание кнопок

Дедушка подвел Дину и Тима к главной задаче сегодняшнего занятия, которую он сформулировал коротко:

— Настало время создать свою собственную кнопку на экране.

Дина высказала мнение:

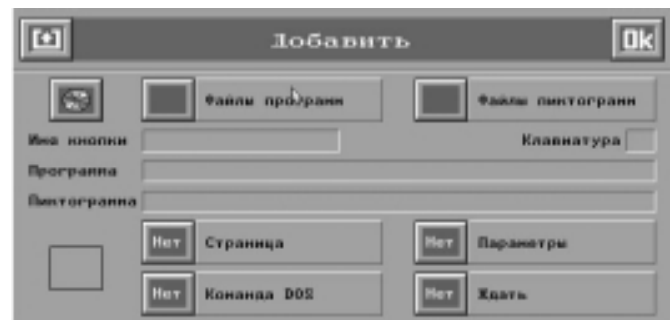
— Я, кажется, догадываюсь, как это сделать. Нужно нажать на "крест" в служебной полоске *Монитора* и в меню выбрать раздел *Добавить*.

Дедушка и на этот раз одобрил идею Дины:

— Точно, Диночка. Давай на страницу *Мудрого Крота* добавим кнопку для вызова программы *Микрон* — это текстовый редактор *Роботландии*, он входит в состав пакета *Конструктор*.

Тим, успевший щелкнуть по красному крестику *Операции*, а затем в позиции *Добавить*, воскликнул:

— Ого! После выбора *Добавить* почти весь экран заняло широкое окно с множеством кнопок и разных полей:



Дедушкины пояснения были необходимы:

— Это окно можно назвать паспортом кнопки. Вы должны его заполнить. Прежде всего, Тим, запиши имя кнопки, то есть тот текст, который будет записываться под кнопкой на экране.

Тим одновременно делал и говорил:

— Сначала устанавливаю курсор в поле *Имя кнопки*. И, конечно, нажимаю левую кнопку мыши. Стрелка заменилась на прямоугольный курсор ввода, и теперь я записываю текст *Микрон*. Выход из поля, я думаю, выполняется по клавише выполнения.

— Здесь для выхода из поля можно нажать на любую кнопку мыши, — сказал дедушка, — или, соответственно,

на любую из клавиш или . Разница лишь в том, хочешь ты или нет сохранить в поле свои изменения.

Тим признал:

— Это удобно. И я понял, когда может выручить клавиша . Если надо немного подправить текст и при этом случайно он будет испорчен, то самое простое — восстановить его клавишей перехода, а потом снова войти в поле и начать редактирование заново.

Дедушка продолжал:

— Теперь в поле *Программа* надо записать вызов программы *Микрон*.

Дедушка отлично понимал, что выполнить это его указание было не так уж и просто. Поэтому свою рекомендацию он подробно и не торопясь развернул:

— Здесь надо указать полный путь к файлу *micron.exe*. Он расположен в каталоге *KURS*, который является подкаталогом *KURSR0B*, расположенного на диске *C:* нашего винчестера (вот тут-то и понадобится нам вторая, подробная схема из параграфа 3 — *Анатомия Монитора*):

`C:\KURSR0B\KURS\micron.exe`

Вызов *Микрона* можно записать и проще, отсчитывая путь от каталога, в котором расположен сам *Монитор*:

`.\KURS\micron.exe`

В этом месте Тим решил, что надо выяснить ситуацию:

— Правильно ли я понял, что обозначение `."` заменяет часть полного пути к *Монитору*, а именно `C:\KURSR0B\`?

Дедушке пришлось уточнять:

— Обозначение `."` указывает текущий каталог. Так как сам *Монитор* запускается из каталога `C:\KURSR0B\`, то этот каталог становится текущим, и все отчеты путей к программам можно вести от него, записывая вместо `C:\KURSR0B\` более кратко: `."`.

Дина заявила:

— Следя за твоими словами, дедушка, мне кажется, что я понимаю. Но объяснить заполнение "кнопкиного" паспорта своим подругам из класса мне, наверное, будет трудно.

— На самом деле, — сказал дедушка обнадеживающе, — вызов программы можно указать гораздо проще, не влезая в такие дебри.

— Как? — в один голос спросили Дина и Тим.

— При помощи кнопки *Файлы программ*.

Тим отметил:

— Но рядом с этим полем находятся две кнопки!

— Используй пока вторую кнопку. Ту, которая расположена ближе к названию, а о первой мы поговорим чуть позже.

Тим последовал дедушкиному совету:

— Нажимаю на эту кнопку. Ага, на экране появилось знакомое изображение каталога диска:



Далее дедушка подсказал:

— Подкаталог *KURS*, в котором расположен *Микрон*, не уместился в окошке. Придется нажать экранную кнопку-стрелку для перемещения окна вниз по списку:



Тим добился нужного представления:

— Вот теперь нужная строка на экране. Щелкаю по ней мышкой и вижу в окне содержимое каталога *KURS*:



Далее дедушка рассказал о понравившемся Тиму способе записи имени файла:

— Теперь ты ищешь в списке файлов строку с *Микроном*, нажимаешь кнопку "Ок" в правом углу окна, и *Монитор* сам прописывает нужный путь в поле *Программа*.

Тим искренне воскликнул:

— Фантастика! Вот это я понимаю — забота о человеке!

А дедушка обратил внимание еще на одну деталь:

— Заметьте, что *Монитор* в качестве пути к *Микрому* записал мой последний вариант с текущим каталогом — `.\KURS\micron.exe`.

Тим и эту подробность сумел оценить:

— Этот вариант короче, и я надеюсь, что *Монитор*-то уж все сделал правильно.

Дедушка открывал ребятам новые и новые возможности каталога *Монитора*:

— Текущий каталог надо использовать не только из-за краткости, но и из-за переносимости.

Дина удивилась неслыханному слову:

— Это что еще такое — "переносимость"?

Дедушка объяснил:

— Представь себе, что ты перенес пакет *Конструктор* с диска *C:* на диск *D:*. Тогда полный вызов *Микрона* — `C:\KURSR0B\KURS\micron.exe` — работать, конечно, не будет, а краткий — `.\KURS\micron.exe` — будет работать по-прежнему.

Этот прием записи пути к файлу ребятам очень понравился, и Тим выразил общее мнение так:

— Верно! Ведь в нем отсчет идет от того места, где находится *Монитор*, следовательно, не важно, где сам *Монитор* расположен.

— Заполняем дальше паспорт кнопки. Не расслабляйтесь, — пригласил дедушка Фёрстов к дальнейшей сосредоточенной работе. — Теперь надо заполнить поле *Пикто-*

грамма, то есть точно таким же образом написать путь к файлу той пиктограммы, которую мы хотим увидеть на кнопке с *Микроном*. Можно, конечно, войти прямо в поле *Пиктограмма* и написать путь вручную...

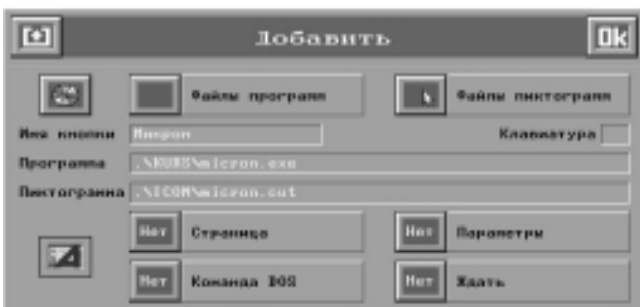
Но Тим, окрыленный последними рекомендациями дедушки, даже не дал ему завершить предложение о “ручном” заполнении поля:

— Нет уж! Я пойду более простым путем. Нажму кнопку *Файлы пиктограмм* и из каталога выберу нужную строку... Только как найти пиктограмму *Микрона*?

— Ну, это очень просто, — сказал дедушка. — Пиктограмма *Микрона* содержится в файле *micron.cut*.

Найти этот файл в каталоге Тиму не составило труда:

— Получилось! *Монитор* записал в поле *Пиктограмма* текст `.\ICON\micron.cut`. Это потому, что все пиктограммы расположены в каталоге *ICON*. Кроме того, изображение пиктограммы появилось в левом нижнем углу нашего “паспорта” — как фотография на настоящем паспорте:



— Что теперь делать дальше? — спросил он.

Дедушка оценил работу внука:

— Практически ты все сделал. Теперь можно нажать “Ок”. Кнопка с *Микроном* появится в рабочем поле, и ты сможешь ее использовать.

Но Тим видел, что не все кнопки в паспорте были использованы, и потому спросил:

— А для чего нужно поле *Клавиатура* и еще четыре кнопки внизу?

— Обычно кнопка на рабочем столе *Монитора* “нажимается” при помощи мыши, но если записать в поле *Клавиатура* какой-нибудь знак, то кнопка на экране будет “нажата” и при помощи клавиши с этим знаком на клавиатуре. Это ускоряет работу с *Монитором*.

— Понятно, — сказал Тим. — Это для Дины. Она у нас большая любительница клавиатуры.

Сестра не обиделась на эту ремарку, но воспользовалась случаем, чтобы сказать:

— Может быть, по такому случаю ты уступишь мне очередь у клавиатуры?

Дина и Тим поменялись местами, а дедушка продолжал, как ни в чем не бывало, словно и не было этой минутной паузы на пересаживание:

— Если вы хотите, чтобы кнопка не вызывала программу, а выполняла переход на новую страницу, надо в поле *Имя кнопки* написать ее название с тремя точками на конце и, нажав кнопку *Страница*, установить на ней слово “Да”. И тогда нет необходимости заполнять поле программы.

Тим отметил и этот этап работы так, словно он, как и несколько минут назад, сидел у клавиатуры:

— И это ясно. Остались еще три кнопки с не очень понятными надписями.

Дедушка сказал еще о двух кнопках:

— Некоторые программы устроены так, что при запуске требуют ввода параметров, то есть величин или имен дополнительных файлов, без которых они не будут работать. Если установить кнопку-переключатель *Параметры* в “Да”, то при запуске программы на экран будет выводиться окно для ввода параметров: компьютер сам попросит их ввести. Потребуется лишь аккуратно выполнить эту просьбу.

При установке переключателя “Ждать” в состояние “Да” возврат из программы в *Монитор* будет происходить не сразу, а только после нажатия на какую-нибудь клавиатурную клавишу.

— А зачем это? — не поняла Дина.

— Ну, предположим, что программа написала на экране текст и закончила свою работу. Тогда если не “ждать”, то ты просто не успеешь этот текст прочитать.

— Понятно... Ну а кнопка *Команда DOS*, вероятно, совсем из “высшего пилотажа”?

Дедушка начал было объяснять:

— Она устанавливается в “Да”, когда запускается не программа, а команда *DOS* (такая, как, например, *dir* или *copy*)...

Но, увидев погрузневшие (от непонимания) глаза Дины, приостановился. К тому же пожаловался и Тим:

— Да... Как я и ожидал, очень умно и не очень понятно. Тогда дедушка сменил тему разговора:

— Предлагаю подробнее об этой кнопке поговорить чуть позже, а сейчас пока двинемся дальше. Хочу только сделать одно замечание, которое вам может пригодиться. Бывают программы, которые при своей работе требуют у компьютера очень много памяти. Случается, что памяти, оставшейся для них после запуска *Монитора*, бывает недостаточно. Для таких приложений в поле *Программа* через один или несколько пробелов от записанного пути помещается символ “#” (дизел). Этот символ предупреждает *Монитор* о “жадной” программе, и он выделяет ей больше памяти.

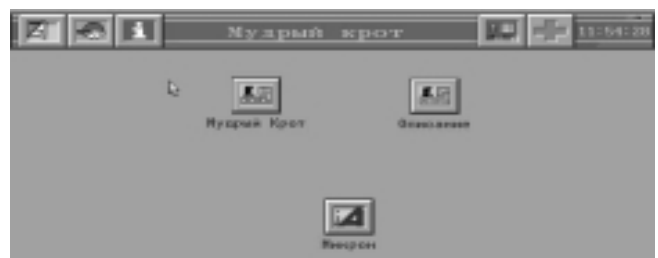
Дина спросила:

— Значит, если программа из *Монитора* почему-либо не запускается, нужно ставить “решетку”?

— Совершенно верно. Как правило, это всегда помогает.

Теперь, когда почти все неясности остались позади, Дина сказала:

— Ну, наконец-то, нажимаю кнопку “Ок” в правом углу окна... и на странице *Крота* действительно появилась кнопка с *Микроном*:



4.5. Сложные вызовы: просмотр файлов

Дина отметила одно отклонение от принятых в *Мониторе* традиций:

— У большинства программ, как я заметила, есть описание. Хотелось бы и рядом с кнопкой запуска *Микрона* иметь кнопку с его описанием.

Дедушка и эту мысль оценил:

— Логичное желание. Для просмотра текстовых файлов в пакете *Конструктор* есть специальная программа `rview.exe`. С помощью нее мы сегодня читали описание *Мудрого Крота*.

— Это ее ты называл “программа-смотрелка”? — спросила Дина.

— Да-да. Это она самая. Запускается программа так — `rview <имя файла>`

Здесь: `rview` — имя программы просмотра, а `<имя файла>` — параметр, имя файла, который надо просмотреть. Обозначение из угольков `< >` используется в описаниях для указания переменных полей, то есть полей, значения которых могут быть разными в каждом конкретном случае.

Тим включился с вопросом:

— Ага, программа просмотра как раз использует те самые параметры, о которых ты так подробно сейчас рассказывал?

Дедушка ответил:

— Всего один параметр — имя того файла, который нужно просмотреть. Понятно, что просто имя указывается лишь тогда, когда текстовый файл и программа `rview` лежат в одном каталоге. Для других случаев надо указывать дисковый путь, так же как мы это делали при задании вызова программы в паспорте кнопки.

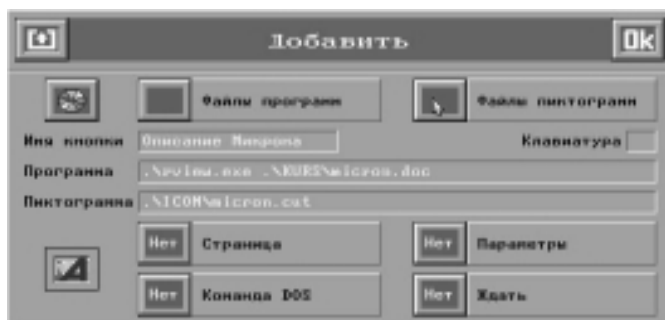
Дина задала уточняющий вопрос:

— Значит ли, что я должна в паспорте кнопки установить переключатель *Параметры* в “Да”?

— Можно и так, но тогда каждый раз при запуске имя файла придется вводить вручную.

— Да, — согласилась Дина, — это очень неудобно.

— Поэтому, — предложил дедушка, — в нашем случае параметр лучше дописать прямо в поле *Программа* на паспорте кнопки.



И он показал, как следует заполнить поле *Программа*, — `.\rview .\KURS\micron.exe`

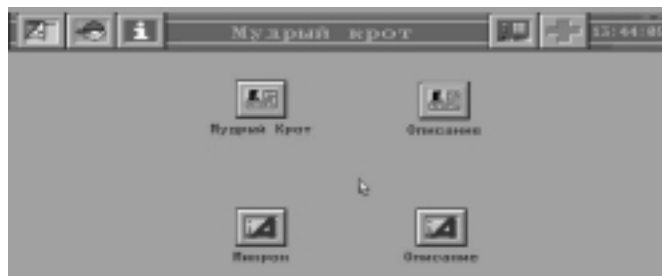
— Первая часть `.\rview` — это вызов программы. Этот текст получился, как обычно, при помощи выбора файла из каталога. Вторую часть — `.\KURS\micron.doc` — параметр — я ввел с помощью первой из двух кнопок, расположенных рядом с полем *Файлы программ*, самой левой.

Дина воскликнула:

— Ага, вот наконец нам пригодилась эта кнопка! А то я уж подумала, что она нарисована просто для украшения...

Дедушка добавил:

— Эта кнопка тоже вызывает *Каталог*, но результат выбора в нем не заменяет текст в поле *Программа*, а дописывается к нему. Очень удобно! Теперь я нажимаю “Ok”, и нужная кнопка появляется на рабочем столе:



Вот вам еще несколько заданий, чтобы поработать с *Монитором* и проверить себя, как вы все поняли:

Задача 2. На странице *Мудрого Крота* поставить кнопки для вызова программы Шарманщик и просмотра его описания.

Задача 3. На странице *Мудрого Крота* поставить кнопку для вывода на экран списка файлов и подкаталогов каталога MOLE.

Задача 4. Перенести кнопку с описанием *Крота* на предшествующую страницу.

4.6. Редактирование кнопок

Договорившись с дедушкой о том, что его задания они с Тимом сделают попозже, на другой день, Дина спросила:

— Помнится, в списке-меню операций *Монитора* был раздел *Изменить*. Для чего он нужен?

— Он нужен для изменения паспорта кнопки. Например, тогда, когда нужно изменить пиктограмму. Вот посмотрите, — принялся рассказывать дедушка (вопрос Дины удачно вписывался в его планы). — Я предлагаю для примера заменить пиктограмму просмотра описания *Микрона* на изображение раскрытой книги. Эта пиктограмма хранится в каталоге ICON под именем `doc.cut`.

— Ну, попробую, — сказала Дина для того, чтобы отслеживать рекомендации дедушки. — Вызываю меню, выбираю раздел *Изменить...* Ага, курсор изменил свою форму и теперь напоминает увеличительное стекло. Что делать дальше?

— Теперь ты должна установить увеличительное стекло на ту кнопку, которую собираешься редактировать, и щелкнуть мышкой.

— Хорошо. Устанавливаю курсор на кнопку *Описание Микрона* и нажимаю клавишу выполнения (тут Дина вспомнила, как брат ехидно отметил ее пристрастие к клавишам; хотела было ему назло щелкнуть левой кнопкой мышки, но привычка взяла верх). На экране появилось такое же окно, как и при создании новой кнопки, но уже заполненное. Оно содержит вызов программы `rview` для просмотра описания *Микрона*.

Дедушка напомнил:

— Тебе нужно изменить пиктограмму.

— Понятно! Щелкаю мышкой по кнопке *Файлы пиктограмм*, нахожу в списке файл `doc.cut`, нажимаю кнопку “Ok”, каталог закрыт и новое описание пиктограммы заменило в поле *Пиктограмма* старое. Теперь нажимаю кнопку “Ok” на панели с паспортом, и на экране — кнопка *Описание Микрона* уже с новым рисунком. Все понятно! Правда, курсор по-прежнему изображен в виде увеличительного стекла.

— Это означает, что ты можешь продолжать редактировать другие кнопки (или ту же самую), не вызывая меню операций.

— А если мне это не требуется? Отказаться от редактирования, как и от удаления, можно клавишей перехода?

Дедушка кивнул:

— Или нажатием на правую кнопку мыши.


Дина продолжила сыпать вопрос за вопросом:

— А что случилось бы, если бы при закрытии панели с паспортом я нажала не кнопку с надписью “Ok”, а другую, изображенную слева, на ней изображена пиктограмма выхода?

Дедушка словно ждал этого вопроса:

— Эта кнопка служит для закрытия окна без сохранения произведенных в нем изменений.

— Значит, если бы я передумала менять пиктограмму, мне не надо было бы восстанавливать имя старой, а просто щелкнуть по кнопке выхода?

— Совершенно верно! Или нажать на клавишу  — результат был бы тем же. Или нажать правую кнопку мыши.

— Это ясно! Клавиша перехода и правая кнопка мыши в *Мониторе* работают одинаково.

Дедушка произнес обобщающую фразу:

— В программах *Роботландии* всегда так: если окно имеет две кнопки для выхода, то правая из них с надписью “Ok” сохраняет изменения в окне, а левая — нет.

Задача 5. Повторить все действия Дины для смены пиктограммы у кнопки *Описание Микрона*.

4.7. Конфигурация

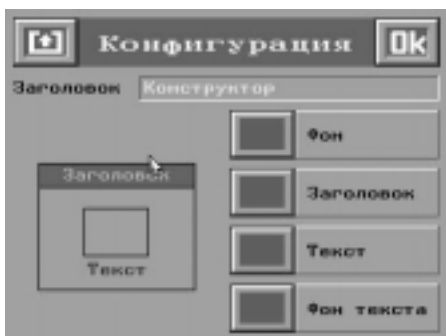
Предвидя недалекое уже окончание занятия, дедушка сказал:

— В меню операций остался еще один раздел, который мы не рассматривали.

Дина догадалась:

— Самый последний. Он называется *Конфигурация*.

— Этот раздел появляется в меню только на самой первой странице *Монитора*. Ну, выбирай его, — предложил дедушка. — Посмотрим, что это такое.



Дине все было ясно:

— Понятно! На этой панели можно изменить название пакета и подобрать цвета для рабочего стола, заголовка и надписей под кнопками.

— Все так. Ну, попробуй, поэкспериментировать.

4.8. Настройки

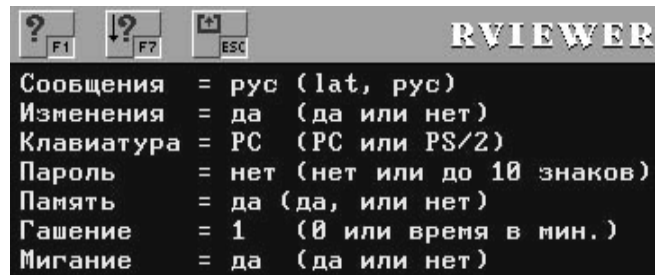
После нескольких минут, ушедших на опыты внуков по перекрашиванию фона рабочего поля и надписей, дедушка вернул занятие в нормальное русло деловой беседы:

— У *Монитора* есть специальный текстовый файл настроек, его имя на диске *desk.ini*.

Тим с интересом спросил:

— А что же можно настраивать при помощи этого файла?

— Давайте посмотрим, как он выглядит, — предложил дедушка Фёров в ответ на этот вопрос своего внука и тут же попросил Дину вызвать этот файл на экран *Микрона*:



Каждая строка в этом файле позволяет устанавливать разные режимы работы *Монитора*.

Тиму пока не все было понятно:

— А как делать изменения?

— Изменения можно выполнять в любом текстовом редакторе, например, в *Микроне*. На рабочем столе страницы *Монитор* в пакете есть специальная кнопка для редактирования файла *desk.ini*:



Дина заявила:

— Щелкая мышкой по кнопке *Настройки Монитора*. На экране появился *Микрон* с файлом настроек.

Дедушка стал рассказывать о файле:

— Первая строка позволяет выбирать язык экранных сообщений. Сейчас в ней после знака “=” и последующего пробела записано слово “рус”. Это означает, что *Монитор* выдает сообщения на русском языке:

Сообщения = рус (lat, рус)

Можно настроить *Монитор* на английский язык, если вместо значения “рус” записать значение “lat”:

Сообщения = lat (lat, рус)

Дина спросила:

— А можно ли настроить *Монитор* на французский язык?

— Можно. Но это уже непросто: нужно сделать специальный заказ фирме *Роботландия*. Наш *Монитор* понимает только два языка — русский и английский.

Вторая строка позволяет запретить редактирование *Монитора*. Если в качестве значения в ней записать “нет”, в *Мониторе* нельзя будет перемещать кнопки, редактировать и удалять их, а также создавать новые:

Изменения = нет (да или нет)

— Я понимаю это так, — объяснила Дина сама себе, — если *Монитор* наполнен кнопками, как надо, то лучше запретить изменения, чтобы случайно в нем что-нибудь не испортить.

Дедушка продолжал беседу о строках INI-файла:

— Третья строка настраивает клавиатуру компьютера.

Тим удивился:

— А разве ее нужно настраивать?

— Обычно нет. Но у стареньких компьютеров IBM PS/2, которые еще встречаются в школах, расположение некоторых надписей на клавишах не соответствует символам, которые появляются на экране. Тогда строка

Клавиатура = PS/2 (PC или PS/2)

поможет восстановить соответствие.

Следующая строка позволяет задавать пароль на выходе из *Монитора*.

Дина усомнилась:

- Ты уверен? Именно на выходе, а не на входе?
- Да. Если, например, в этой строке написать

Пароль = робот (нет или до 10 знаков),

то выйти из *Монитора* смогут только те пользователи, которые знают это волшебное слово.

— А для чего же запрещать выход из *Монитора*?

— А представь себе, что ты оставила за компьютером своего друга — новичка в информатике, а сама пошла в магазин за сухариками. Если пароль задан, то ты не будешь нервничать в очереди. Твой друг не сможет, случайно покинув *Монитор*, что-нибудь натворить на винчестере компьютера.

Дина усомнилась:

- Он может посмотреть пароль в настроечном файле.
- Ну, до этого еще нужно догадаться. А потом кнопку редактирования файла настроек можно вообще убрать с рабочего стола *Монитора*.

Следующая строка выручает, когда возникают проблемы с памятью у приложений, запускаемых из-под *Монитора*. Установка

Память = да (да или нет)

равнозначна установке “решетки” в паспорте каждой кнопки (см. п. 4.4). Этот параметр удобно использовать, если в пакете много “жадных” до компьютерной памяти приложений.

Тим предложил такое решение:

— Так, может, сразу сделать такую установку и больше никогда не мучиться?

— Если так сделать, то запуск программ из *Монитора* будет выполняться немного медленнее. Для очень “слабых” машин это может оказаться существенным.

Следующая строка позволяет установить время в минутах, через которое будет происходить гашение экрана, если пользователь не нажимает на клавиатуре клавиши и не работает с мышью.

Дина спросила:

— А это для чего предусмотрено?

— Для удлинения срока службы дисплея компьютера. Ведь когда он ярко светится, электронный луч выжигает люминесцентное покрытие экрана.

Наконец, последняя строка позволяет убрать мигание кнопок-страниц на рабочем столе *Монитора* тем пользователям, кого это раздражает.

Тим сказал:

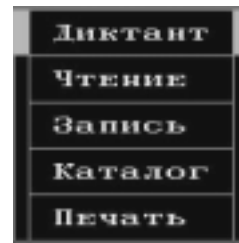
— А мне мигание нравится. Сразу видно, что кнопка не запускает программу, а выполняет переход на другую страницу.

Дедушка против этого не возражал:

— Я тоже не против мигания кнопок, но люди такие разные...

На вопрос Дины: “А как записать сделанные изменения в настроечном файле?” — дедушка дал простой ответ:

— Так, как обычно, это делается в *Микроне*. Сначала клавишей перехода вызывается меню:



Потом выбирается раздел *Запись*. На экран выдается панель запроса имени, но вводить имя не надо, оно там уже написано, просто нажимается **Enter**:



Теперь на экране появляется окно-предупреждение:



Устанавливаешь метку на “Да” и нажимаешь **Enter**. Новая редакция файла настроек записана. Теперь остается только нажать аккорд **Ctrl** + **Break** для выхода из *Микрона* и возврата в *Монитор*.

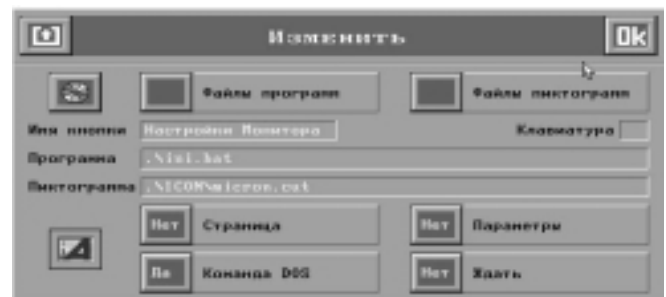
4.9. Сложные вызовы: редактирование файлов

Тим заявил:

— Меня заинтересовал способ вызова *Микрона* с файлом для редактирования. Как это устроено?

— Давай посмотрим паспорт кнопки *Настройка Монитора*, — предложил дедушка и попросил Дину:

— Дина, выбери в меню операций *Монитора* раздел *Изменить* и курсор-лупу установи на эту кнопку. Так. Нажми на **Enter**.



Дина, выполнив просьбу дедушки, сказала:

— Здесь видно, что кнопка запускает файл `ini.bat`.

— Файл с расширением `BAT` (такие файлы часто называют просто `bat`-файлами) составляет из команд операционной системы. Поэтому в окне *Команда DOS* установлено “Да”. Это специальный командный файл, он расположен там же, где и файл настроек `desk.ini` на диске. Вот содержимое файла `ini.bat`:

```
.\kurs\micron -- desk.ini
```

Первая часть `.\kurs\micron` — это вызов программы *Микрон*, расположенной в каталоге `KURS`, а дальше располагаются три параметра, из которых первые два пропущены (на их место поставлены черточки-тире). Третий параметр *Микрона* как раз и задает имя того файла, который будет загружен в *Микрон* при его запуске.

Дина уточнила:

— Значит, если мне надо в *Мониторе* сделать кнопку для редактирования текстового файла, я должна в ее паспорте написать вызов ВАТ-файла, а сам ВАТ-файл поместить в тот каталог, в котором расположен редактируемый файл?

— Совершенно верно, — сказал дедушка. — А ВАТ-файл ты должна построить, взяв за образец ВАТ-файл, редактирующий файл настроек *Монитора*.

Задача 6. На странице *Мудрого Крота* поместить кнопку для редактирования файла с его описанием.

5. Изготовление пиктограмм

Тим задал вопрос, который уже давно напрашивался на сегодняшнем занятии:

— Я хотел бы нарисовать свою собственную пиктограмму и поместить ее на кнопку. Как это сделать?

— Как поместить пиктограмму на кнопку, ты уже знаешь: нужно записать путь к ней в паспорте кнопки. А для рисования можно использовать *Редактор Пиктограмм*. Он вызывается предпоследней кнопкой в служебной полоске *Монитора*.

— Вижу. На этой кнопке изображен карандаш... — сказал Тим и снова обратился к Дине: — Щелкни, пожалуйста, мышкой по *Редактору Пиктограмм*.



Дети с интересом посмотрели на окно редактора, и дедушка вновь взял слово:

— Пиктограммы рисуют под 36-кратным увеличением при помощи таких инструментов (вот они на средней панели редактора):

- точка;
- отрезок (ломаная);
- прямоугольник (эллипс);
- закрашенный прямоугольник (закрашенный эллипс).

В инструментальном меню — только 4 кнопки с изображением инструментов, в то время как инструментов — 7. Щелчком правой кнопки мыши на изображении инструмента можно поменять его на другой. Кроме инструмента “точка”.

Самый удобный инструмент для управления этим редактором — мышь. Впрочем, как всегда, можно работать и с клавиатуры. При этом клавиша **Enter**, как всегда, соответствует левой кнопке мыши, а клавиша **Esc** — правой.

Палитра

При помощи палитры устанавливаются два цвета — “верхний” и “нижний”. Верхний цвет выбирается левой кнопкой мыши, нижний — правой. Это правило действует и при рисовании: при нажатой левой кнопке работает верхний цвет, при нажатой правой — нижний. Верхний и нижние цвета изображаются на кисти, нарисованной на экране в центре палитры.

Тим перебил дедушкин рассказ:

— Я думаю, что рисовать двумя цветами очень удобно. Например, один цвет можно выбрать для контура рисунка, а другой, цвета фона, для стирания.

— Это верно, — согласился дедушка. — Теперь я расскажу о том, как пользоваться инструментами.

Точка

Одиночный щелчок оставляет на поле точку верхнего цвета, если нажата левая кнопка мыши, и точку нижнего цвета, если нажата правая кнопка. Перемещение мыши с удерживаемой кнопкой позволяет превратить инструмент в тонкую кисть.

Отрезок

Инструментом можно рисовать как отдельные точки, так и отрезки прямых при помощи “резиновой нити”. Использование левой или правой кнопки мыши приводит к рисованию соответственно верхним или нижним цветом. Алгоритм рисования отрезков такой:

- поиск подходящего места для начала отрезка (перемещение мыши);
- фиксирование начала (нажатие и удерживание кнопки мыши);
- поиск подходящего места для конца отрезка (перемещение мыши с нажатой кнопкой);
- фиксирование линии на поле (отпускание кнопки).

Ломаная

Ломаную линию рисуют по такому алгоритму:

- поиск подходящего места для начала ломаной (перемещение мыши);
- фиксирование начала ломаной (щелчок кнопкой мыши);
- пока нет щелчка другой кнопкой мыши:
 - поиск подходящего места для конца отрезка (перемещение мыши);
 - фиксирование отрезка на поле (щелчок кнопкой мыши).

Прямоугольники и эллипсы

Алгоритм рисования этими инструментами ничем не отличается от алгоритма рисования отрезком, только, конечно, результат рисования будет другим.

Дина повторила уже высказанную на сегодняшнем занятии мысль:

— Когда слушаешь тебя, вроде все понятно, но когда начинаешь что-то делать сам, возникают вопросы.

— Нужна небольшая практика, — ободряюще сказал дедушка, — редактор очень простой и удобный, ты быстро привыкнешь. На чем я остановился? — спросил он, чуть сбитый со своей линии повествования неожиданным вопросом.


Тим сказал:

— Ты говорил об инструментах в редакторе пиктограмм. Но, кажется, ты сказал уже обо всех инструментах. Расскажи нам теперь про кнопки, расположенные в нижней части окна редактора.

— Вот эти кнопки:

Залить все — закраска нижним цветом целиком всей пиктограммы.

Заменить цвет — точки, окрашенные в нижний цвет, красятся в верхний во всей пиктограмме.

Еще в редакторе можно выполнять перемещения рисунка так, как будто листочек, на котором он нарисован, склеен наподобие шара. Эта операция называется циклическим сдвигом и выполняется стрелками клавиатуры при удерживаемой клавише .

— Удобная вещь! — признала Дина. — Можно переместить рисунок точно в центр. А как быть с ошибками, которые я допускаю при рисовании?

— Во-первых, ты можешь один цвет использовать для стирания, а во-вторых, есть откатка и накатка рисующих операций.







— Эти операции выполняются, как обычно, по-роботландски?

— Да:

Откатка —  + 

Накатка —  + 

Про чтение и запись пиктограмм я рассказывать не буду — они делаются обычным способом, как в *Микроне*.

Выход из редактора выполняется при помощи экранной кнопки “STOP” или клавиатурных аккордов:  + ,  +  или  + .

— Ну а кнопка с вопросительным знаком “?” в служебной полоске напомнит нам то, что я забуду из твоего рассказа, — это наверняка вызов подсказки, — без труда догадался Тим, показав не обсуждавшуюся пока кнопку.

— Верно, — сказал дедушка. — А теперь я советую вам немного поупражняться. Вот вам несколько заданий:

Задача 7. Нарисуйте пиктограммы: красный кружок на голубом фоне, в середине цифра 1.

Задача 8. Нарисуйте пиктограмму *Стрелка вправо* в нижней части рабочего поля *Редактора пиктограмм*. Поднимите стрелку в центр рабочего поля.

Задача 9. Нарисуйте пиктограмму: на зеленом рабочем поле голубой квадрат, окаймленный желтой рамочкой. Воспользуйтесь для этого позициями *Закрашенный прямоугольник* и *Отрезок* в меню *Редактора*.

6. Сборка нового пакета

Два из этих упражнений (восьмое и девятое) ребята решили отложить на самостоятельную работу следующего дня, а седьмое задание выполнили в присутствии дедушки и под его одобрительные замечания.

Впрочем, и сами ребята были довольны — теперь они могли сами создавать пиктограммы и располагать их на экране *Монитора*.

Дина высказала такое образное сравнение:

— *Монитор* не ставит ограничения для пользователя, а, наоборот, широко открывает ему двери, чтобы в этом доме пользователь мог похозяйничать.

— Хорошее сравнение, — заметил дедушка. — Недаром *Монитор* относят к открытым программам.

— Что ты понимаешь под этими словами — “открытая программа”? — не понял Тим.

— Примерно то, о чем сказала сейчас твоя сестра: в *Мониторе* можно многое менять и настраивать. Например, при желании можно изменить его служебные пиктограммы. Они, как и пиктограммы пользователя, располагаются в каталоге ICON.

— А какие имена у служебных пиктограмм? — попробовал уточнить Тим.

— Все они начинаются с обозначения *sys* (и дедушка показал перечень этих пиктограмм в каталоге):

sys00.cut — STOP
 sys01.cut — Отогнутый листок
 sys02.cut — Фирменный знак Роботландии
 sys03.cut — Владелец пакета
 sys04.cut — Редактор Пиктограмм
 sys05.cut — Меню операций
 sys06.cut — Выход без сохранения
 sys07.cut — “Ок”
 sys08.cut — Стрелка вверх
 sys09.cut — Стрелка вниз
 sys10.cut — Добавить имя файла

Тим заявил:

— Изменить пиктограммы я могу, конечно, в *Редакторе Пиктограмм*. Но я не буду этого делать. Мне нравятся рисунки на служебных кнопках, и сомнительно, чтобы я нарисовал их лучше.

Дедушке понравилась скромность заявления внука:

— Но тем не менее такая возможность у тебя остается, — и, посчитав тему конструирования пиктограмм законченной, он сказал: — А теперь я предлагаю собрать новый пакет, управляемый *Монитором*.

— А что мы включим в этот новый пакет? — с живым интересом откликнулась Дина.

— Пусть на первой странице *Монитора* будут две кнопки:



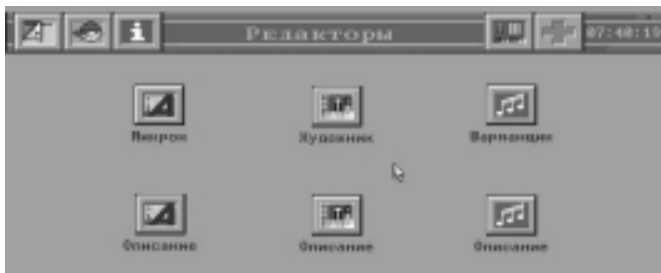
Тим рассудительно заметил:

— Так как на твоём рисунке стоят многоточия после имен кнопок, то, я думаю, это будут две новых страницы.

— Правильно. Давайте на странице *Игры* поместим *Мудрого Крота* и программу *Привет* с их описаниями, а на странице *Редакторы* — программы *Микрон*, *Художник* и *Шарманщик*.

— Тогда эти страницы должны выглядеть так, — заявила Дина после того, как, дружно и усердно потрудившись вместе с Тимом, она показала, как выглядят изготовленные страницы:





Дедушка признал задание выполненным:

— У меня нет возражений. Давайте расположим новый пакет на диске C: в каталоге ПАКЕТ.

Тим возразил:

— Но такого каталога нет на нашем диске.

Дедушка не считал это возражение препятствием для дальнейшей работы:

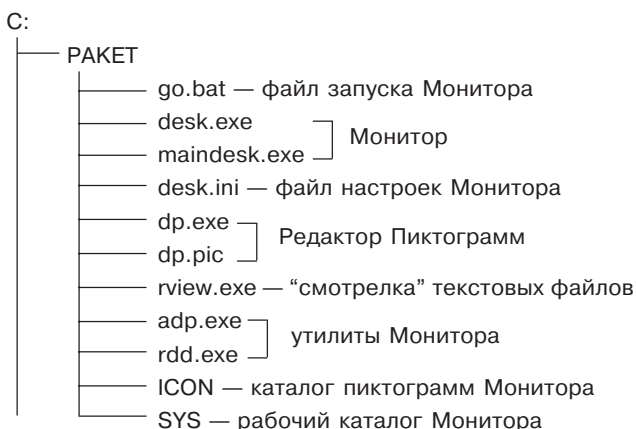
— Ну, пустой каталог ПАКЕТ я легко создам в любой оболочке — и в DOS, и в Windows. Так, в Norton Commander нужно всего лишь нажать клавишу **F7**, а в Windows, находясь в окне диска C:, нажать правую кнопку мыши и в выпавшем окошке-меню выбрать команду *Создать папку*. Не забудьте только и в том, и в другом случае ввести с клавиатуры имя нового каталога-папки.

Выполнив необходимые операции, Дина сообщила:

— Каталог на диске появился, но он пустой. Что делать дальше?

— Сначала в него нужно скопировать из каталога KURSROB все, что относится к *Монитору*, а именно все файлы, расположенные в этом каталоге, кроме файла *ini.bat*, и подкаталоги *SYS* и *ICON*.

Эту работу я тоже могу с одинаковым успехом сделать и в Norton Commander, и в Windows. Вот что у меня получилось:



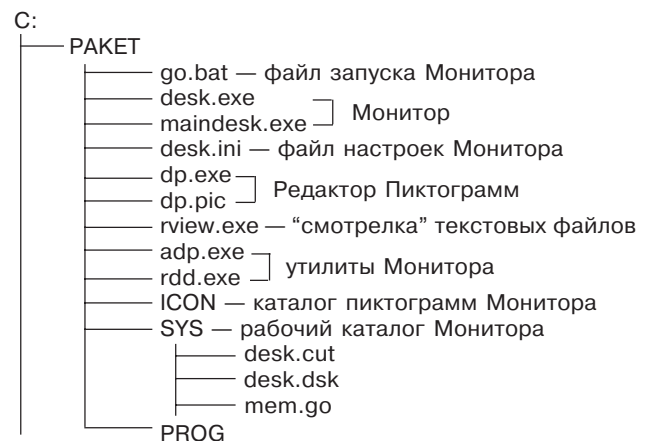
Теперь надо войти в каталог *SYS* и удалить там все файлы *deskp.000*, *deskp.001*, *deskp.002*, ... — это описание страниц пакета *Конструктор*, они нам здесь не нужны. В каталоге *SYS* нужно оставить только три вспомогательных файла *Монитора*:

```

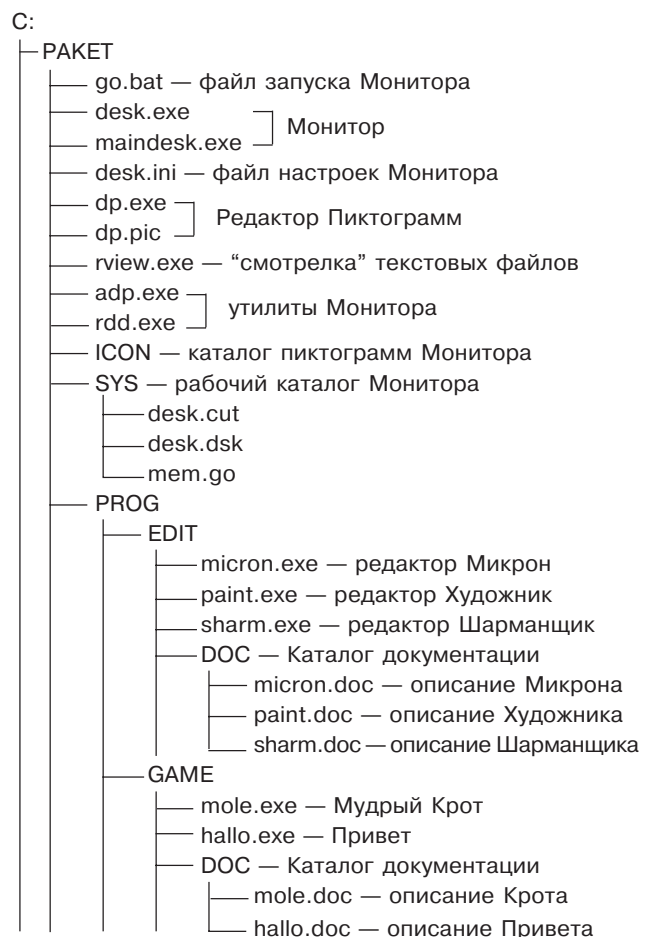
desk.cut
desk.dsk
mem.go

```

Далее, давайте создадим пустой рабочий каталог, в который будем копировать программы нашего пакета. Назовем его *PROG*. Сделано. Теперь схема файлов пакета выглядит так:



Дина и Тим посоветовались и предложили дедушке такую схему расположения файлов программ и документации в каталоге *PROG*:



Дедушка выразил удовлетворение:

— Ну что ж, схема получилась неплохой. Теперь в соответствии с ней можно выполнить всю техническую работу: создать дополнительные каталоги *EDIT*, *GAME* и два подкаталога *DOC*, а также скопировать туда нужные файлы из пакета *Конструктор*. Готово. Теперь надо заняться обустройством самого *Монитора*.

Задача 10. Запустить *Монитор* командой *go* в новом каталоге *ПАКЕТ* и создать в нем кнопки в соответствии со схемами дедушки Фёрстова.

Задача 11. Придумать схему другого пакета, составленного не обязательно из программ *Роботландии*, и собрать его на винчестере.

7. Утилиты Монитора

7.1. Монитор на приеме у доктора

Тим заявил:

— Новый пакет с играми и редакторами собран, но в каталоге ICON осталось много лишних пиктограмм. Хотелось бы избавиться от этого ненужного “балласта”.

У дедушки Фёрстова всегда найдется решение:

— В поставочном комплекте *Монитора* есть две весьма полезные программы: rdd.exe и adp.exe. Первая из них поможет решить твою задачу.

— А как мне ее использовать?

— Просто запусти на выполнение файл rdd.exe.

Тим, выпросив к этому времени у сестры разрешение снова сменить ее у компьютера, сказал:

— Я это сделал, и на экране появились разные сообщения, а также и странный вопрос:

```
RDD (Robotland DeskDoctor) ver 1.0 12-05-95
(C) Robotland
В Мониторе нет обращений к пиктограмме "pic1.cut".
Что делать?
Удалить пиктограмму      - 1
Не удалять                - 2
Удалять все такие        - 3
Не удалять пиктограммы  - 4
```

Это меня спрашивают, что делать?

— Ну, конечно, тебя. Программа RDD используется для проверки *Монитора* и его служебных файлов. Сейчас она обнаружила, что пиктограмма pic1.cut в *Мониторе* лишняя, и спрашивает тебя, что делать. Для ответа ты должен нажать клавишу с соответствующей цифрой.

— Понятно. Лишняя пиктограмма мне не нужна, поэтому я нажимаю клавишу с цифрой 1. Теперь на экране появились новые сообщения и старый вопрос:

```
Пиктограмма "pic1.cut" удалена.
В Мониторе нет обращений к пиктограмме
"pic2.cut".
Что делать?
Удалить пиктограмму      - 1
Не удалять                - 2
Удалять все такие        - 3
Не удалять пиктограммы  - 4
```

Дедушке нетрудно было и здесь дать то же пояснение:

— Программа спрашивает, удалять ли ей пиктограмму pic2.cut. Я советую тебе нажать клавишу с цифрой 3. Тогда все оставшиеся лишние пиктограммы будут удалены без всяких вопросов.

Тиму идея понравилась:

— Так я и сделаю. На экране появился целый список удаленных пиктограмм и сообщение:

Число обнаруженных ошибок в пакете - 0.

Дедушка сказал:

— Лишние пиктограммы не считаются ошибками. Я советую тебе запускать программу rdd.exe время от времени и обязательно каждый раз, когда в работе *Монитора* обнаружатся какие-нибудь странности.

— Я понял. Эта программа делает *Монитору* врачебный осмотр и выполняет необходимое лечение.

7.2. Сложение Мониторов

В конце занятия, несмотря на заметную усталость внуков, дедушка предложил:

— Нам осталось рассмотреть еще один важный вопрос о сложении *Мониторов*, и тогда Фёрстовы станут большими мониторными специалистами.

Дина вставила суждение:

— Ну вот добрались до арифметики...

Но дедушка поправил ее:

— Я имею в виду задачу об объединении двух пакетов, работающих под *Мониторами*, в один общий пакет. Ставлю задачу более конкретно. Пусть на нашем диске C: имеются два пакета. Вот структура первого пакета:

```
C:
├── ПАКЕТ1
│   ├── go.bat — файл запуска Монитора
│   ├── desk.exe
│   └── maindesk.exe ] Монитор
│   ├── desk.ini — файл настроек Монитора
│   ├── dp.exe
│   └── dp.pic ] Редактор Пиктограмм
│   ├── rview.exe — “смотрелка” текстовых файлов
│   ├── adp.exe
│   └── rdd.exe ] утилиты Монитора
│   ├── DOC — Документация по Монитору
│   ├── ICON — каталог пиктограмм Монитора
│   └── SYS — рабочий каталог Монитора
│       ├── desk.cut
│       ├── desk.dsk
│       └── mem.go
└── PROG1 — каталог программ, запускаемых в Мониторе
```

А вот так выглядит второй:

```
C:
├── ПАКЕТ2
│   ├── go.bat — файл запуска Монитора
│   ├── desk.exe
│   └── maindesk.exe ] Монитор
│   ├── desk.ini — файл настроек Монитора
│   ├── dp.exe
│   └── dp.pic ] Редактор Пиктограмм
│   ├── rview.exe — “смотрелка” текстовых файлов
│   ├── adp.exe
│   └── rdd.exe ] утилиты Монитора
│   ├── DOC — Документация по Монитору
│   ├── ICON — каталог пиктограмм Монитора
│   └── SYS — рабочий каталог Монитора
│       ├── desk.cut
│       ├── desk.dsk
│       └── mem.go
└── PROG2 — каталог с программами,
           которые запускаются в Мониторе
```

Требуется добавить второй пакет к первому, так чтобы первый пакет принял вид:

```

C:
├── ПАКЕТ1
│   ├── go.bat — файл запуска Монитора
│   ├── desk.exe
│   ├── maindesk.exe ] Монитор
│   ├── desk.ini — файл настроек Монитора
│   ├── dp.exe
│   ├── dp.pic ] Редактор Пиктограмм
│   ├── rview.exe — “смотрелка” текстовых файлов
│   ├── adp.exe
│   ├── rdd.exe ] утилиты Монитора
│   ├── DOC — Документация по Монитору
│   ├── ICON — каталог пиктограмм Монитора
│   ├── SYS — рабочий каталог Монитора
│   ├── PROG1 ] каталоги программ, запускаемых из
│   └── PROG2 ] общего Монитора

```

Решение задачи распадается на две части. Сначала нужно скопировать каталог PROG2 в каталог ПАКЕТ1. Затем нужно к страницам *Монитора* первого пакета добавить страницы *Монитора* из второго пакета, а это уже сделать простым копированием в Norton или Windows нельзя. Кроме того, нужно перенести в каталог ICON первого пакета недостающие пиктограммы из второго, но ничего вручную делать не следует. Нужно воспользоваться вспомогательной программой `adp.exe`, которая расположена в первом пакете.

— Так что же надо делать? — растерялся Тим, посмотрел на сестру, но и та смотрела на задачу не очень весело.

Дедушка стал рассказывать алгоритм решения задачи:

— Сначала скопировать каталог PROG2 в каталог ПАКЕТ1, затем войти в каталог ПАКЕТ1 и набрать в командной строке следующую команду:

```
adp C:\ПАКЕТ1 C:\ПАКЕТ2\
```

— Что означает эта запись? — спросила Дина.

— `adp` — это имя запускаемой программы, а за именем идут два параметра. Первый параметр задает путь к пакету, в который идет добавление, а второй — путь к пакету, который добавляется. Так как текущий каталог сейчас `C:\ПАКЕТ1`, то вызов программы `adp` можно записать проще:

```
adp .\ C:\ПАКЕТ2\
```

— И что же произойдет? — захотел узнать Тим.

— Программа `adp.exe` поставит на первую страницу пакета ПАКЕТ1 новую кнопку, которая будет открывать страницу с кнопками второго пакета. Затем она переносит в первый пакет всю структуру страниц из второго пакета вместе с кнопками и нужными пиктограммами. Иными словами, после запуска `adp` можно сразу начинать работать, ничего больше не настраивая.

Нужно, правда, будет подобрать или нарисовать пиктограмму для новой кнопки и, быть может, перенести эту кнопку на другую страницу.

Хочу еще заметить, что в документации по *Монитору* программы `rdd.exe` и `adp.exe` называются утилитами. Так всегда называют небольшие вспомогательные программы.

Задача 12. Создать на диске два новых пакета, управляемых *Мониторами*, и, пользуясь утилитой `adp.exe`, объединить их в один общий пакет.

8. Кое-что из упущенного

— Мы посмотрели главные выполняемые *Монитором* операции. Но не все.

Тим улыбнулся:

— *Монитор* еще может чистить картошку?

8.1. Временный выход из Монитора

Оценив шутку Тима, дедушка сказал:

— Нет, но из *Монитора* можно временно выйти в командную строку DOS, нажав клавишу **Break** на клавиатуре.

— А зачем это надо?

— Ты можешь выполнить команду `dir, copy` или вызвать Norton и что-то сделать в нем, а потом вернуться в *Монитор* на то самое место, из которого его покинул.

— И что же нужно сделать для возврата?

— В командной строке набрать приказ `exit`.

8.2. Запуск предваряющего приложения

— А что еще мы пропустили? — спросила Дина.

— Еще можно предварять запуск *Монитора* каким-то другим приложением. Фирма *Роботландия* так часто делает в своих пакетах, предваряя запуск *Монитора* запуском программы-заставки с мультипликацией и музыкальным оформлением.

— А это как делается? — Дине эта возможность *Монитора* показалась интересной.

— Очень просто, — ответил дедушка и, в свою очередь, задал вопрос: — Как ты запускаешь *Монитор*?

Дина ответила:

— Набираю команду `go`.

— И что при этом происходит?

Дина удивилась вопросу:

— Как что? Запускается *Монитор*, и я в нем работаю.

Дедушка хотел услышать другое:

— Дина, я не про это... Когда ты набираешь команду `go`, запускается командный файл `go.bat`, он расположен там же, где файлы самого *Монитора*. Сейчас я выведу этот файл на экран, и ты увидишь, что в нем написано: одно слово — `maindesk`.

Дина прочитала надпись, хоть она и была сделана по-английски:

```
— maindesk.
```

— Это означает, — сказал дедушка, — что командный файл `go.bat` запускает программу `maindesk.exe` — это ядро *Монитора*. Если ты отредактируешь файл и запишешь в нем `maindesk demo.exe`,

то теперь запуск программы `demo.exe` будет предварять запуск самого Монитора.

— Кажется, я поняла, — удовлетворенно сказала Дина.

9. Конкурс пиктограмм

Свою следующую встречу с внуками дедушка Фёрстов начал торжественной тирадой:

— Сегодня вас ожидает творческая работа на всероссийском конкурсе пиктограмм.

Дина и Тим не всегда умели различить, когда дедушка говорит серьезно, а когда шутит. Сейчас они немного смущались, потому что про такой большой конкурс они до сих пор не слышали.

Увидев замешательство ребят, он сменил свой пафосный тон:

— Вы напрасно удивляетесь. Ведь все наши занятия — это ваше участие в Роботландском сетевом университете, где, кроме вас, учатся тысячи ребят из всей России, и не только из России. Сейчас мы закончили большую тему, и куратор нашего курса предлагает вам, то есть нашей команде, так же, как и всем другим командам, проверить свои знания.

— Проверить как? — спросила Дина. — К нам придет профессор из университета и начнет задавать хитрые вопросы?

— Ну, конечно, нет, — рассмеялся дедушка. — Самое лучшее средство проверить новые знания ученика — попросить его сконструировать новый продукт — программу, картинку, текст — так, чтобы при таком конструировании можно было применить новые знания. Мы с вами находимся именно в таком положении: вы научились создавать пиктограммы, вот куратор и просит вас показать, как вы это делаете. А раз строить пиктограммы будут многие — все те, кто учится на курсе “Компьютерное конструирование” Роботландского университета, то пиктограмм получится много, и, значит, интересно сравнить их, устроив конкурс.

Дина робко спросила:

— А ты нам можешь, дедушка?

— Я подскажу вам, если вы забудете, какие клавиши надо нажимать. Помогу отправить ваши сообщения и вложить в них рисунки пиктограмм. Но все самое главное — придумать пиктограммы, построить рисунок, подобрать подходящие краски, даже описать назначение пиктограммы — должны вы сами. Это ведь творческий конкурс, в котором проверяется, как вы умеете конструировать пиктограммы.

— Вот ты сказал, “придумать пиктограмму”. Но ведь мы должны знать, на какую тему придумывать, — попытался уточнить Тим.

— Об этом я вам скажу, потому что условия конкурса, конечно, приведены в письмах, которые куратор курса разослал всем руководителям команд. И мне в том числе. Я готов вас познакомить с этими условиями.

И, вызвав на экран электронное письмо, которое пришло из Роботландского университета, он прочитал:

“... для участия в конкурсе команда должна выслать в Университет комплект пиктограмм для Монитора, разработанных детьми по теме “Школьные предметы и их разделы”.

Ниже приводится неполный список примеров по некоторым подтемам:

География (физическая география частей света, флора и фауна земного шара и стран СНГ, полезные ископаемые России, государственные флаги, ...)

История (древняя история — Египет, Месопотамия, Греция, Рим; история России; династии русских царей; Великая Отечественная война, ...)

Литература (русские писатели XIX века, литература советского периода, литературные жанры, зарубежная литература, ...).

Пока дедушка сделал короткую паузу, Тим успел шепнуть Дине:

— Ничего себе — набор тем! Это же про все на свете. Да еще назвали как — “примеры по некоторым подтемам”, а в конце каждой подтемы еще и многоточия. Мол, додумывайте сами! Проще было сказать — выбирайте любую тему — не ошибетесь.

Дедушка не стал оспаривать эту мысль, а продолжал читать:

“Математика (разделы математики, основные понятия школьного курса геометрии, элементы теории множеств...)

Музыкальное воспитание (элементарная музыкальная грамота, биографии великих композиторов, выдающиеся музыкальные творения, ...)

Физкультура и спорт (гигиена и здоровье, олимпийское движение, эмблемы видов спорта, ...) ...

Пиктограммы могут предназначаться как для обозначения целых разделов (страниц Монитора), так и для вызова приложений с задачами по конкретным темам. Набор пиктограмм должен сопровождаться текстовым файлом с описанием каждой из них по следующей схеме:”

Тут успела вставить свой комментарий Дина:

— Вот здесь слушай внимательно: надо будет не только рисовать, но и писать, сочинять тексты!

“Заголовок файла

- руководитель команды: фамилия, имя и отчество
 - регистрационный номер команды
 - e-mail команды
- Описание пиктограмм**
- название пиктограммы, предлагаемое автором
 - назначение
 - имя файла
 - автор (авторы), фамилия и имя

Пиктограммы (файлы с расширением СUT) и текстовый файл с их описанием (с расширением ТХТ) следует послать по адресу: <pervin@robot.botik.ru> по сети в Университет до 3 ноября (включительно”).

Тим снова остановил дедушку:

— Давайте разберемся. Это нас с Диной так называют — “авторы”?

— Ну конечно. Вы же создаете новый рисунок — пиктограммы, который будет не похож ни на какой ранее созданный, да к тому же еще и дадите его описание. Это настоящая самостоятельная, авторская, творческая работа.

Ребята постепенно начали осознавать ответственность предстоящей работы, но и одновременно ее интерес.

У Дины появился еще один вопрос:

— На всяком конкурсе есть судьи. Их называют жюри. А кто будет оценивать конкурсные пиктограммы? Да и как их оценивать? Кто скажет, что в пиктограмме хорошо, а что — плохо?

— Ты, как всегда, задаешь интересные вопросы, Дина, — ответил дедушка. — И много. На все сразу я ответить не смогу...

Дина согласно кивнула. А дедушка выбрал вопрос для ответа:

— Важнее начать с последнего: каковы критерии в оценке пиктограммы. Или, как ты сказала, “Что такое хорошо и что такое плохо в пиктограмме”. За каждую пиктограмму ставится не одна оценка, а пять разных. Вот они (и дедуш-

ка снова обратился к экрану, читая письмо с условиями конкурса):

- **Практическая значимость пиктограммы:**
если пиктограмма может быть рекомендована к практическому использованию в качестве метки какой-либо используемой в школе компьютерной программы, существующей, потенциально реализуемой или актуальной, — то работа оценивается по этому критерию баллами 8—10 (10 — это максимальная оценка), для того чтобы претендовать на такую высокую оценку, применимость должна быть указана автором пиктограммы в ее описании;
если описание отсутствует, но легко определяется практическим предназначением, пиктограмма получает оценку 5—7;
если применимость пиктограммы не ясна или вызывает сомнения, пиктограмма получает оценку 2—4.
- **Смысловая выразительность:**
максимальный балл по этому критерию — 8; балл 6—8 получает пиктограмма, однозначно определяющая то явление, процесс или предмет, которое должна помечать пиктограмма; если соответствие пиктограммы и помечаемого объекта неоднозначно, пиктограмма получает 4—5 баллов;
если соответствие, по мнению эксперта, отсутствует или противоречит замыслу, описанному автором, результат оценивается баллом 1—3.
- **Художественное (графическое) представление пиктограммы:**
максимальное количество баллов — 8; в интервале от 1 до 8 эксперту предлагается оценить цветовую гамму рисунка-пиктограммы и ее композицию.
- **Оригинальность:**
самостоятельно придуманные и реализованные пиктограммы получают высший балл — 4; если пиктограмма близка по замыслу к одной из пиктограмм, используемых на панелях пакета Конструктор, или несколько модифицирована, ее оценивают баллом 2—3; полностью скопированная пиктограмма оценивается баллом 0.
- **Наличие описания:**
максимальная оценка — 6 баллов;
хорошее нестандартное описание — 6;
краткое точное описание — 4 балла;
неточное или ошибочное описание — 2 балла;
отсутствие описания — 0 баллов.

В описании надо указать фактические параметры пиктограммы — имя, название, автор (фамилия, имя, класс), а также назначение пиктограммы (в каких предметах и темах, для обозначения каких объектов) возможно ее использование. Обоснование выбора сюжета — наиболее красивая и весомая часть описания. Стихотворные описания вполне допустимы”.

Дина нетерпеливо предложила:

— Мы сделаем несколько пиктограмм. И Тим нарисует. И я. А может быть, мы сделаем какие-нибудь пиктограммы вместе.

Тим согласился:

- Не беспокойся, дедушка, мы не подеремся.
- Я вам верю, — сказал довольный дедушка. — А потом мы с вами обсудим результаты вашего творчества, выберем лучшие пиктограммы и пошлем их в университет.
- А сколько нам дается времени?
- Совсем немного, — огорчил дедушка ребят. — В вашем распоряжении всего три вечера: в субботу надо отправлять пиктограммы по электронной почте. Но, к счастью, в эти дни я не буду занимать компьютер надолго, потому что ко мне приезжает мой давний товарищ, которому я должен буду показать город, рассказать о нашей школе. Да мало ли, о чем нам предстоит поговорить. Мы так давно не виделись. Так что вы уж тут хозяйничайте сами. Кстати, условия конкурса я оставляю на столе: я их уже успел распечатать на принтере.

Понимая, какое ответственное дело им предстоит, ребята тут же взялись за работу. Сначала они придумали сюжеты пиктограмм. Посоветовались друг с другом. Увидели, что в их задумках есть и общие темы (ведь они оба с интересом относились, например, к информатике) и сразу решили, какие из пиктограмм будут рисовать вместе.

— Дедушка, ведь у одной пиктограммы могут быть два автора? — решила удостовериться Дина.

— Бессспорно. Правилами это не запрещено.

10. Перекрестная проверка

Когда через три дня команда снова собралась вместе и приняла единое решение о посылаемых на роботландский конкурс пиктограммах, дедушка отправил коллекцию творений своих внуков по электронной почте. Он написал несколько строк о своей команде, о выборе отправляемых пиктограмм, о том, как ребята трудились над рисунками и их описаниями. При этом дедушка по ходу дела показывал ребятам, как работает почтовая программа, с помощью которой он готовил отправление этой коллекции. Наконец, он щелкнул по кнопке *Отправить письмо*.

— А вот теперь, — сказал дедушка, обращаясь к Дине и Тиму, — самое время поговорить о вопросе Дины, который остался без ответа: кто проверит эти пиктограммы?

— Да, конечно, — в один голос согласились дети. — Это нам очень интересно.

Дедушка начал говорить, и дети удивленно переглянулись. Они думали, что дедушка ответит одним словом, попросту назвав фамилию проверяющего, а рассказ обещал быть длинным и подробным:

— Для проверки работ в университете назначается специальная методическая комиссия. В нее входят несколько человек — три или четыре. Это люди, хорошо знающие предмет и умеющие решать проверяемые задачи. Их называют экспертами. Каждый эксперт проверяет задачу отдельно, не советуясь друг с другом.

Таким образом, у каждой задачи получается сразу несколько оценок...

— Это как в фигурном катании, — подсказала Дина, которая не пропускала ни одной телевизионной трансляции с соревнований по фигурному катанию. (В этом “виновата” бабушка: она в молодости была увлечена фигурным катанием и сейчас стремилась зажечь этим увлечением любимую внучку). — Там судьи из судейской коллегии выставляют оценки точно так же.

— И на КВН тоже, — вспомнил Тим еще одну популярную телепередачу.

— Верно, — оценил дедушка удачные параллели. — Тогда вы подскажите мне, что с этими судейскими оценками происходит дальше.

— А дальше, — за двоих ответила Дина, — оценки всех судей... (хм, — она тут же поправилась, — “экспертов”) надо сложить и получить средний балл, то есть разделить полученную сумму на число экспертов.

— Очень хорошо, — подтвердил дедушка точно выраженной мыслью. — Правда, оценка получается не целой, а дробной. Но зато она гораздо точнее выражает качество оцениваемого фигуристами танца или выступления в КВН, а в нашем случае — качество пиктограммы. А самое главное в такой экспертной системе оценивания — объективность, то есть независимость от вкусов и предпочтений отдельных судей-экспертов. Действительно, если у какого-нибудь судьи и оказался “любимчик”, то за счет того, что эксперты сдают в судейскую коллегию (или по-нашему — в методическую комиссию) несколько независимых оценок, единое мнение коллегии-комиссии выравнивается и не выглядит однобоким.

— Но ведь ты нам, дедушка, еще в прошлый раз рассказывал, что за пиктограмму ставится не одна оценка, а пять. Значит, эксперты по каждой пиктограмме выводят пять средних оценок? — спросил Тим.

— Так тоже можно было бы действовать. Но этот способ работы неудобен. Поэтому методическая комиссия поступает по-другому. Для каждой пиктограммы, как ты заметил, Тим, эксперт выставляет пять оценок, но в методическую комиссию выдает только одну — сводную. По этой общей оценке и происходит сравнение оцениваемых пиктограмм.

Вот посмотрите, с какими таблицами имеют дело эксперты (и он для примера показал с экрана одну из экспертных таблиц):

Эксперт Кошкин

Автор (авторы)	К-да	Название пиктограммы	Имя файла	O1	O2	O3	O4	O5	OO
Петров Вова	2107	Слон	slo.cut	9	6	7	3	5	25
Иванова Света	2120	Космос	kos.cut	10	6	6	4	4	30
Сидоров Коля	2112	Музыка	not.cut	9	5	6	4	5	29

У Дины эта таблица вызвала несколько вопросов:

— Все-таки нужны пояснения. Что тут написано в заголовке таблицы? Хитрое слово “к-да” — это, видимо, сокращение от непоместившегося слова “команда”? А вот что означают нолики в названиях последних шести столбцов?

Дедушка пояснил:

— С номером, то есть обозначением команды, ты разобралась правильно. Номер команды необходим в этой таблице потому, что наш конкурс не только личный, но и командный тоже. А значки, которые ты назвала “ноликами”, вовсе не нолики, а буквы “O”:

- O1 — первая оценка (то есть практическая полезность),
- O2 — вторая оценка (смысловая выразительность),
- O3 — третья оценка (художественное представление),
- O4 — четвертая оценка (оригинальность),
- O5 — пятая оценка (наличие описания).

— Значит, “OO” в последней графе — это...

— Общая оценка! — догадался Тим.

Дедушка признал его правоту и продолжал:

— Как вы догадываетесь, у каждого эксперта на столе стоит компьютер. Поэтому и таблицы они заполняют не на бумажных листочках, а на компьютерных экранах. И общую оценку эксперты подсчитывают тоже не на бумаге и даже не на калькуляторе, а на том же самом компьютере. Такой удобный инструмент надо использовать при каждой возможности.

Когда все общие оценки таким образом получены, то и упорядочить строки таблицы по убыванию величины OO компьютер тоже сможет. И вот вам готовый результат конкурса — таблица, где в первой строке стоит лучшая пиктограмма, а все остальные упорядочены в порядке уменьшения столь непросто подсчитываемого качества.

Тим заметил одну недосказанность дедушки:

— Теперь ясно, как по такой таблице можно подвести итоги личного первенства участников. И то, если я правильно понимаю, это не конкурс участников-авторов, а конкурс пиктограмм: ведь автор может выступать на конкурсе с несколькими работами, и тогда одна из них вполне может оказаться в верхних строчках таблицы, а другая, похуже, — внизу...

— Ты рассуждаешь очень верно, — оценил дедушка Фёрстов, перебивая внука.

Но Тим еще не закончил свой вопрос:

— А ведь ты говорил, что одновременно проводятся два конкурса — личный и командный. Как же подводятся итоги командного первенства?

— Коль скоро мы обращаемся за помощью к компьютеру, то и в этом вопросе он нам поможет. Действительно, коль скоро в каждой строке есть номер команды, то по этому значению компьютер способен выбрать все строки с задаваемым номером в отдельную, таким же образом устроенную таблицу. В такой таблице собраны все работы одной команды. Теперь остается только сложить в ней все общие оценки OO и разделить на количество представленных работ, то есть на число строк в таблице команды. Вот так и получается сводная командная оценка. А уж расставить два или три десятка команд по значению единственного признака — это для компьютера не работа, а пустяк.

— Здорово! — удовлетворенно признала Дина. — Мне очень нравится такая работа методической комиссии, потому что в этой работе компьютер и люди трудятся вместе, четко разделив между собой обязанности. От такой комиссии не стыдно получить даже не очень хорошую оценку. Ясно будет, что справедливо.

Казалось, правила оценивания пиктограмм были уже растолкованы, но ребята заметили, что дедушка хочет продолжить разговор на эту тему, и приготовились его слушать.

— В действительности в итогах конкурса учитываются не только мнения экспертов, но и участников.

Дина и Тим переглянулись (“Что-то новенькое! Трудно себе представить, чтобы в классе ученики сами себе выставляли оценки за контрольную по математике!”).

Дедушка продолжал:

— Прежде всего куратор очень рекомендует перед отправлением пиктограмм в университет сделать их самопроверку.

— То есть мы сами себе поставим оценки по тем правилам, о которых ты нам рассказал? — задала удивленно вопрос Дина.

— Именно это я и хотел сказать.

— Но тогда где же... как ты ее назвал... объективность? Ведь каждому хочется получить оценку получше.

— Верно. Но точно так же каждому автору хочется, чтобы его мнение о собственной работе совпадало с мнением окружающих его людей. Поэтому я уверен, что вы сможете критически отнестись и к своим собственным работам. Кстати, такая самопроверка поможет вам в еще одном важном задании, которым завершается конкурс пиктограмм и, как вы увидите позднее, вообще всякий конкурс в Роботландском университете.

Ребята навострили уши (“Что же дедушка имеет в виду, говоря о последнем задании в конкурсе?”).

— После получения от команд всех конкурсных работ куратору курса предстоит выполнить два серьезных дела. Первое из них мы уже увидели — нагрузить работой методическую комиссию. Второе (более сложное и более интересное) — отправить пакеты с работами команд в другие команды курса.

— Зачем? — не удержались от вопроса и Тим, и Дина.

— Для перекрестных проверок, — ответил дедушка, но, понимая, что сам по себе новый термин детям ничего не объясняет, продолжил: — В таком задании, как перекрестная проверка, проверяющая команда должна проверить все работы команды, которые куратор прислал на проверку. Следовательно, в перекрестной проверке каждая проверяющая команда выступает как коллективный эксперт. Тем самым объективность сводной оценки повышается еще больше, потому что работа по подведению итогов, о которой мы только что беседовали, на самом деле включает не только результаты экспертов из методической комиссии университета, но и результаты перекрестных проверок команд. Таких результатов может оказаться немало. Насколько я знаю, наш куратор организует работу так, что каждой команде (нашей в том числе) он отправляет на проверку пакеты трех или даже четырех команд. Поэтому вам предстоит немало потрудиться с проверкой пиктограмм своих однокурсников. В этой работе самое главное — быть объективным, то есть строгим, но справедливым. Тогда все будут уверены, что сводные оценки по курсовому конкурсу будут объективными. Это значит — честными и правильными.

Дети с интересом прослушали этот дедушкин монолог. Идея перекрестных проверок им действительно понравилась. Вопросы последовали немедленно:

— А мы такие задания на проверку получим? А когда они к нам придут? Сколько команд нам придется проверять?

Дедушка замахал руками:

— Нельзя столько вопросов задавать сразу. Все рассказу, не спешите. В университете существует строгий график всех работ. По этому графику завтра мы получим работы на перекрестную проверку. Предлагаю вам сегодняшний вечер потратить на самопроверку.

Дина тут же предложила:

— Мы можем к тому же потренироваться в “перекрестной проверке” — я поставлю оценки за пиктограммы Тиму, а он — мне.

— Предложение принимается? — спросил дедушка Тима; тот согласно кивнул. — Отвечаю на следующий вопрос. В последнем письме, которое пришло из университе-

та, написана просьба — проверить пиктограммы трех команд. Справимся?

— Конечно! — ответил дружный дуэт.

— Наконец, последний вопрос: работы каких команд мы будем проверять? На этот вопрос я вам ответить не могу. Более того, сейчас ответ на этот вопрос не знает даже куратор.

Эта последняя дедушкина фраза была непонятна:

— Как же так? А что же он тогда будет отправлять?

— Он будет посылать те пакеты, которые ему посоветует компьютер. Ведь план отправления писем на проверку составляет не человек, а компьютер, который старательно и внимательно проверяет, чтобы проверяемые работы на проверку не попали в ту команду, в которой пиктограммы были созданы.

Ребята загорелись желанием поскорее посмотреть на пиктограммы своих однокурсников из других городов и сел:

— Это очень хорошая идея — перекрестная проверка! Да и полезная: мы не только научились делать пиктограммы, но увидим, как это делают другие ребята. Слово продолжение урока, — оценила Дина значение предстоящей важной работы.

11. Итоги

Прошли два дня увлеченной работы по перекрестной проверке. Дина и Тим с удовольствием обнаружили, что созданные ими рисунки ничуть не хуже многих других работ, какие им пришлось проверять. Впрочем, и пользу они извлекли несомненную. Так, “сухие” описания пиктограмм у Тима и даже у Дины заметно уступали, например, работам учащихся из сельской школы села Филимоново, где описания были написаны не только точно, но и красиво — в стихотворной форме.

Еще два дня пришлось ждать письмо из университета. Это было подробное письмо, которое, конечно, предстояло прочесть: ведь в этом письме анализировались успехи детей в усвоении важной темы — проектировании пиктограмм. Но можно было понять ребят, которые упростили дедушку бросить взгляд в первую очередь на итоговые таблицы.

Дедушка вывел на экран документ, вложенный в только что полученное электронное письмо. Это была таблица, которая, как и следовало ожидать, не убралась на экране целиком (как объяснил дедушка, на конкурс были присланы несколько сотен пиктограмм).

Таблица итогов личного первенства конкурса пиктограмм начиналась так:

11lit0	31.67	Поройков Артем
19krab	30.50	Филиппов Филя
10diti	30.25	Фёрстовы Дина и Тим
12mont	30.00	Ракова Саша, Клятченко Сергей
11um0	30.00	Ледкова Ю., Мартынов М.
11idda	30.00	Диков Д., Берсенев А.
10dika	30.00	Фёрстова Дина
24lad0	30.00	Павлов Виталий
10timt	29.80	Фёрстов Тим
19kr20	29.80	Кайдалова Даша
25gur1	29.80	Журавкова Лена
01lalo	29.67	Кустова Клава
11saf0	29.50	Сафронов Егор

— Сейчас мы повнимательнее посмотрим всю таблицу, но начало очень-очень приятное. — От удовольствия дедушка радостно потирал руки. — Я поздравляю вас, мои юные коллеги (Дина и Тим понимающе подмигнули друг другу: дедушкины торжественные речи — это признак его хорошего настроения). Если во множестве из сотен работ три пиктограммы нашей команды вошли в первую десятку, это о многом говорит. Поздравляю вас!

Команда	Итоговый балл
2111	34.8
2110	34.6
2102	34.5
2101	34.3
2122	34.3
2125	34.1
2118	34.0
2106	34.0
2109	32.9
2120	32.8
2117	32.8
2112	32.7
2116	32.6
2113	32.5
2124	31.1
2102	31.9
2108	31.9
2107	31.9
2114	31.5
2103	31.4
2105	31.4
2115	31.2
2119	31.2
2121	30.8
2123	30.6

— И мы тебя тоже, — ответила благодарная Дина за двоих, — ведь про конструирование пиктограмм мы все узнали у тебя.

— Ладно, ладно, — умиротворенно произнес дедушка Фёрстов. — А давайте посмотрим, как выглядит таблица командного пер-

венства. У нас должны быть неплохие результаты. И он вывел на экран таблицу (см. слева) итогов командного первенства конкурса пиктограмм.

— Второе место! — воскликнул Тим. — Да за такие успехи на Олимпийских играх дают серебряные медали.

— Ну, серебра для вас я не найду. А вот торт в честь такого успеха бабушка вам сегодня испечет.

— Ура!!! — закричали дети так громко, что дедушка шутливо зажал уши, а бабушка обеспокоенно заглянула в комнату:

— Что тут у вас случилось?

Дедушка ей объяснил:

— Это они радуются твоему будущему торту.

А обращаясь к детям:

— Пока торт не готов, я предлагаю посмотреть картинку с лучшими пиктограммами конкурса. Тем более что среди них — три наши (см. ниже).

КОНКУРС ПИКТОГРАММ

ИНФОРМАТИКА



Газета "Информатика" и Роботландский сетевой университет продолжают совместную акцию

"Подписчикам везде у нас дорога... и скидка!"

Данный купон дает право на скидку в размере 10% при приобретении:

- интерактивного учебника-лаборатории "Знакомство с компьютером";
- программного пакета "Конструирование";
- интерактивного учебника-лаборатории "HTML-конструирование" (см. № 21, 22/2000);
- программного пакета "Зимние вечера" (см. № 1, 5—11, 13—18/2001);
- интерактивного учебника-лаборатории "Javascript-конструирование" (см. № 21, 25, 29, 33/2001).

Для получения скидки необходимо выслать заявку на приобретение того или иного продукта по адресу: 152025, г. Переславль-Залесский, ул. Октябрьская, д. 43, кв. 112, Дуванову Александру Александровичу.

В письмо необходимо вложить оригинал данного купона или ксерокопию купона вместе с ксерокопией подписной квитанции на "Информатику". Для быстрого получения программ рекомендуется дополнительно отправить электронное письмо с заявкой по адресу: kurs@robotland.botik.ru. В электронном письме требуется указать дату отправки бумажного письма с купоном или ксерокопиями.

Данное предложение действительно до 30 апреля 2002 г.



Книга для учителя

Вводная глава

I. В качестве введения

Два обстоятельства привели авторов к идее написания этой книги.

Первое связано с деятельностью Ротландского сетевого университета, который читателям “Информатики” нет необходимости представлять: публикации университета часто появляются на страницах газеты. Один из курсов этого дистанционного обучающего центра — “Компьютерное конструирование” — ставится для слушателей университета уже не первый учебный год и пользуется популярностью как у школьников, так и у педагогов. Методические материалы этого курса — учебник для школьника и пособие для учителя, руководителя команды, — легли в основу книги.

Содержание обучения на курсе вплотную связано с прикладными программными системами, да и технология общения учащихся с куратором курса требует адекватного программного обеспечения. Поэтому читателю-учителю будет полезно при знакомстве с теми или иными методическими приемами или описываемыми в книге упражнениями обратиться непосредственно к компьютерному экрану, запустить соответствующую программу. Поскольку одна из целей, которые ставятся перед читателями-учителями, — конструирование программных объектов, то такой настойчивый учитель должен, как нам представляется, самостоятельно создать по крайней мере один такой объект по каждой изучаемой теме.

Курс “Компьютерное конструирование” в Ротландском сетевом университете предназначается учащимся 5—6-х классов. Следовательно, та же ориентация на читательский контингент сохраняется и у учебника.

Вторым обстоятельством можно было бы назвать публикацию в апрельских номерах “Информатики” концепции школьного курса информатики, предложенной А.А. Кузнецовым, А.Л. Семеновым и А.Ю. Уваровым, а также завязавшейся сразу после ее опубликования дискуссии. Если говорить точнее, хотелось предложить некоторые конструктивные аргументы в такую дискуссию, показав, что кажущиеся экстраординарными методические положения не только могут быть реализованы, но сегодня реализованы уже. И концепция, и дискуссия представляют интерес уже потому, что в публикациях нашли отражения и принципиальные административные решения — варианты учебных планов для предстоящего (пока экспериментально) двенадцатилетнего обучения в школе.

Один из этих учебных планов (увы, только один из трех!) включает в себя информатику как отдельный школьный предмет (и этим он особенно интересен читателям “Информатики”). На первый взгляд предлагаемая реализация информатического школьного образования кажется парадоксальной: сначала уроки информатики в начальной школе, затем “информатическая” ниша центральных классов средней школы и, наконец, возвращение к информатике в старших, выпускных классах. Мы не будем останавливаться сейчас на аргументах авторов концепции, тем более что основные ее положения, и в частности “парадоксальная” структура непрерывного школьного курса информатики, были чуть более простыми словами выска-

заны еще в далеком 1979 году А.П. Ершовым, Г.А. Звенигородским и Ю.П. Первиным — “Школьная информатика (состояние, концепции, перспективы)”, Новосибирск, ВЦ СО АН СССР, 1979 (более поздняя и доступная перепечатка: “Информатика и образование” № 1, 1995). Важнее то, что такой подход имеет конструктивную реализацию. Правда, предложенная Ротландским университетом реализация родилась и пока существует в рамках дистанционного обучения. Однако мы хорошо ощущаем актуальность важнейшего в сегодняшней педагогике лозунга: “Дистанционное обучение — от дополнительного образования к базовому!”. Более того, школьные учителя, являющиеся слушателями университета, постоянно обращаются с просьбами (а иногда с требованиями) дать привязку методических материалов тех или иных занятий к реальным, базовым школьным урокам.

Больше всего таких запросов у слушателей курса “Компьютерное конструирование” (случайно ли?!).

В рамках обсуждаемой концепции информатика не изучается как самостоятельная дисциплина в классах, начинающих среднюю школу, — пятом, шестом, седьмом. Вместе с тем учебный план насыщен предметными компьютеризованными уроками по всем предметам — математика и физика, история и русский язык, химия и иностранный, физкультура и черчение. Для этого необходимы новые методики, а также необходимый инструментарий. Этим проблемам и посвящена предлагаемая на обсуждение книга.

Школьник занимается наполнением программных оболочек и конструированием программных объектов, которые имеют прямой выход на школьные предметные уроки, а учитель, участвующий со своей командой в курсовых турнирах и конкурсах, знает, что одним из важнейших критериев в оценке сконструированного его командой продукта станет практическая значимость: дети не только изучают свои разработки для использования на предметных уроках, но и описывают (своими словами, слегка подкорректированными руководителем команды — учителем) методику их учебного применения.

Активным участием пяти-, шестиклассников в такой дидактически важной деятельности мы хотели показать:

— во-первых, обоснованность положения, по которому учащиеся младших классов средней школы занимаются не фундаментальными понятиями и механизмами информатики в рамках отдельного самостоятельного предмета, а изучают прикладные, технологические аспекты самых широких применений компьютеров в рамках повседневной деятельности человека (в наиболее близкой ему, с точки зрения учащегося, учебной деятельности);

— во-вторых, к такого рода учебной деятельности (где сочетаются мировоззренческие и технологические аспекты информатики) учащиеся пятых классов оказываются подготовленными тем содержанием обучения, которое предусматривается базовым планом для начальной школы; именно такая начальная информатическая подготовка была вынесена в предшествующие курсы раннего обучения информатике в Ротландском университете. Это дало основание подойти к подготовке этой книги как к продолжению публикации “Зимние вечера. Информатика для начинающих”, прошедшей в газете “Информатика” всего год назад.

Тем самым связь с “Зимними вечерами” — это не столько продолжение сюжетной линии и единство персонажей, сколько обоснованная дидактическая линия непрерывного школьного информатического образования.

II. Общие замечания и рекомендации

Программное обеспечение курса — это программные оболочки, являющиеся интегрированными системами, наполняя которые по своему усмотрению, учащийся (или коллектив, команда) создает самостоятельный продукт, имеющий реальное практическое применение. Таковы *Конструктор сказок*, программы *Собеседник*, *Клеточка*, *Тропинка*, *Волшебные* и *Веселые картинки*.

Названные интегрированные среды Роботландии включают в себя возможности изготовления текстовых, музыкальных и графических фрагментов и сборки из них законченного компьютерного произведения. Такие среды являются эффективным инструментарием для организации обучения детей работе на компьютере.

Во-первых, обучение, с точки зрения ученика, уходит на второй план, являясь для него не мотивом, а как бы побочным эффектом деятельности. Основной мотив — создание законченного компьютерного приложения, имеющего практическую ценность (компьютерная сказка, обучающая программа, справочник, тест, игра).

Во-вторых, ученик овладевает компьютерным инструментарием в комплексе, параллельно осваивая разные способы создания компьютерных объектов, имеет возможность проводить не отсроченные по времени аналогии, сравнивая типы создаваемых информационных объектов, интерфейсы программ, методы проектирования.

В-третьих, конечный продукт, созданный детьми в интегрированной среде конструирования, приближен по внешнему виду к профессиональным продуктам. Это достигается переключением на систему основных операций по созданию интерфейса проектируемого приложения. Таким образом, ребенок получает внешне привлекательный продукт, не отвлекаясь от поставленной задачи на технические детали и затрачивая на работу минимальное время.

В-четвертых, ученик во время работы осваивает технологии проектирования сложных систем. За основу, например, может быть взята технология “сверху вниз” (метод постепенной детализации).

В-пятых, появляется возможность естественным образом моделировать работу детей в творческих коллективах, работающих над одним большим проектом.

III. Как работать с учебником

Руководителю команды в Роботландском сетевом университете, как правило школьному учителю, не предлагаются никакие жесткие схемы проведения занятий со школьниками. Более того, разработка конкретного плана и методики работы с группой может стать предметом творчества учителя.

Книга рассказывает о годовом курсе “Компьютерное конструирование”, реализуемом в Роботландском сетевом университете и рассчитанном главным образом на учащихся пятых и шестых классов.

Занятия (желательно двухчасовые) планируются один раз в неделю. С учетом каникул — осенних, зимних и весенних — и несколько сокращенного учебного года (с

середины октября по начало мая), по сравнению с традиционным школьным, получается 22—25 занятий. Эти занятия несколько условно распределены на два семестра, а внутри семестров еще и по разделам-темам.

В каждом занятии учителю важно выделить содержание, методику (цели занятия, его план, способы контроля, комментарии и обсуждения), а также программное сопровождение урока.

Руководитель команды теоретически имеет три варианта предъявления детям учебной информации.

Первый из них — это распечатка текстов занятий из учебника в достаточном количестве экземпляров и распределение их между детьми для последующего обсуждения.

Второй вариант — совместное прочтение учебного материала с экрана компьютера или с его твердой копии.

Наконец, третий вариант — это подробный разговор учителя с детьми на тему занятия из учебника. При этом речь может (и должна!) идти не о прямом пересказе диалогов дедушки Фёрстова с его внуками, а о творчески подготовленной беседе со своими учащимися на заданную учебником тему с использованием примеров из учебника, а также заданий и упражнений, встроенных в каждую главу-тему.

При такой форме занятия текст учебника может рассматриваться как дополнительный, раздаточный материал, если у учителя есть возможности (техника, ресурсы, время) для распечатки страниц учебника. В этом случае выданные ученикам страницы можно эффективно использовать для закрепления материала темы и для домашних заданий. Тем ученикам, которые имеют доступ к домашним компьютерам, можно переписать файлы изученных на уроках заданий, с тем чтобы ребенок смог дома перечитать текст, просмотреть картинки, выполнить упражнения.

В этом последнем случае учитель должен быть уверен, что домашний куратор поможет ученику в технической обработке учебного материала — чтении с дискеты, разархивировании файлов, выводу их на экран, антивирусной проверке, установке необходимых параметров чтения с экрана (выбор и размещение файла, установка масштаба изображения и проч.). Даже если учебная информация из школьного кабинета информатики на домашний компьютер передается не на дискете, а по электронной почте, требования к подготовленности домашнего куратора по приему и предварительной обработке полученной информации остаются.

Готовясь к очередному занятию, учитель должен сам поработать с каждой запускаемой на уроке программой во всех ее режимах, отметить особенности их работы. Такой учительской самоподготовке может помочь пользовательская документация, которая для многих программ раскрывается щелчком по пиктограмме с названием *Описание*. В файл описания программы легко войти с рабочего стола, поскольку запускаемая программа и ее описание расположены, как правило, по соседству и имеют одинаковые пиктограммы.

В ходе каждого занятия предусматривается работа детей на компьютере. Идеально, если каждому ребенку может быть предоставлено индивидуальное рабочее место. Но, даже работая в парах, каждый ребенок должен унести с занятия сформированные или закрепленные навыки управления программой.

Программы пакета *Конструктор* в соответствии с традициями Роботландии выполнены как открытые программы. Это значит, что они могут рассматриваться как программные оболочки, которые наполняются обновляемыми к каждому уроку данными.

Спецификой дистанционных курсов являются контролируемые занятия, в которых взаимная (перекрестная) проверка становится неотъемлемой частью учебно-воспитательного процесса. Именно здесь проявляются столь характерные для дистанционного обучения дисциплинарные требования к срокам, синхронизации и оформлению. Учитель должен приложить немало усилий не только для того, чтобы его ученики решили задачу, но и для того, чтобы они оформили свою работу в строгом соответствии с заданиями. В частности, во многих задачах контролируемых занятий — соревнований в не меньшей степени, чем собственно верное решение, оценивается обоснование найденного решения. Важно, чтобы дети решили и оформили все задачи полностью самостоятельно. Конечно, учитель может и должен помочь ребенку в исправлении грубых орфографических ошибок или даже в клавиатурном наборе (впрочем, трудно поверить в то, что ребенок уступит учителю, даже уважаемому, право набрать на клавишах сочиненный им текст), но цель занятий — научить ребенка делать, а не сделать за него.

Из иницилирующего письма, направленного куратором курса “Компьютерное конструирование” слушателям Роботландского сетевого университета:

“...Занятия в Роботландском сетевом университете, к которым вы сегодня приступаете, имеют некоторые особенности, не присущие урокам и контрольным работам за школьной партой или лекциям и семинарам на вузовской скамье. За предстоящий год вы, вполне вероятно, не увидите лицом к лицу с вашими преподавателями и не услышите их голоса (хотя принципиально и обмен фотографиями, и голосовое общение сегодня реализуемы в компьютерных сетях). У вас, как правило, не планируются личные встречи с вашими однокурсниками и у ваших учеников — с их сверстниками.

И тем не менее с этими людьми, разными по возрасту, семейному положению, опыту работы и многим другим характеристикам, вам предстоит познакомиться, причем, мы надеемся, познакомиться настолько близко, чтобы доверять свои откровенные мысли коллективному обсуждению проблем — методических, организационных, технических и программных вопросов курса раннего обучения информатике в школе — своим новым и пока незнакомым коллегам: преподавателям университета и однокурсникам из десятков городов и сел, мнение которых вы будете регулярно читать на своих экранах.

В таких необычных условиях нам важно с самого начала определиться в педагогических позициях, выбрать единую, общую базу, на которой будут строиться наши отношения. В сегодняшнем, первом сообщении, направляемом вам, уважаемые слушатели Роботландского университета, хотелось бы обсудить гипотезу по структуре таких отношений, чтобы оперативно обмениваться мнениями по этому вопросу и, возможно, внести уточнения в наши планы.

Итак, априорно мы предполагаем ориентироваться на следующие исходные установки, касающиеся предварительной подготовки слушателей.

1. Организационно-целевые установки

Слушатель нашего отделения — это, как правило, школьный учитель, который собрал вокруг себя группу молодых людей примерно 5—6-х классов, увлеченных компьютером и его замечательными возможностями. Ваши занятия с этой группой — кружок, факультатив, дополнительные занятия... (В ваших ближайших письмах найдите, пожалуйста, возможность написать мне о том, какую организационную форму для своих занятий вы выбрали, как вы ее называли, какой график установили, есть ли у вас помощники из старшеклассников, родителей, других учителей, как отнеслось к вашей идее участия в Роботландском университете ваше руководство — директор школы, руководитель Департамента образования; заодно — о комплекте используемых вами программ.)

2. Программно-технические установки

У слушателей не предполагаются умения программировать на каком-либо языке программирования. Это означает, что учитель-предметник не должен бояться первых шагов в нашем курсе, даже если он и не преподавал информатику. Вместе с тем общее представление об информатике, ее основных понятиях и инструментах вполне логично предполагать у нынешнего школьного учителя. Более того, тот факт, что учитель подал заявление на университетский курс, где в центре внимания — методика раннего обучения информатике, свидетельствует, на наш взгляд, о том, что мы имеем дело с единомышленниками в оценке роли информатики как стержневой дисциплины общего образования.

В нашем курсе используются в основном программы из программно-методической системы “Роботландия” или ее расширения — “Хиты Роботландии”. Они собраны в курсовой пакет “Конструктор”, который является основным дидактическим инструментом курса.

3. Инструментальные (сетевые) установки

Основным инструментом общения с университетскими преподавателями и однокурсниками станет для вас электронная почта. Не исключаемые, вообще говоря, традиционные способы общения мы предлагаем использовать только в исключительных ситуациях: междугородний телефон слишком дорог, а обычная почта слишком медлительна.

Коль скоро все заявки пришли в университет по электронной почте, можно было бы считать, что эти заявки присланы людьми, свободно владеющими коммуникационными технологиями. Тем не менее, если у вас возникают вопросы, связанные с правилами работы в информационной сети, вы смело, не смущаясь, можете задавать их вашим однокурсникам и куратору; они не останутся без ответов...”

Глава 1. Монитор

Монитор представляет собой графическую оболочку, назначение которой — запускать приложения (другие программы или команды DOS). На экране *Монитора*, как на странице, нарисованы кнопки. Одни из них запускают программы, другие открывают страницы с новыми кнопками. Управляется *Монитор* при помощи клавиатуры и мыши.

Работая с *Монитором*, пользователь может:

- предварить запуск *Монитора* запуском приложения;
- настроить *Монитор* (язык сообщений, тип клавиатуры, запрет на модификацию страниц и кнопок, пароль на выходе);
- создать новую кнопку;
- удалить кнопку;
- изменить кнопку;
- изменить конфигурацию страницы;
- нарисовать новую пиктограмму или исправить старую.

При создании кнопки пользователь может:

- перенести описание одной кнопки в другое;
- задать имя кнопки;
- задать запускаемое приложение;
- указать на повышенное требование приложения к памяти;
- задать пиктограмму;
- задать соответствие между экранной и клавиатурной кнопками;
- установить следующие переключатели:
 - страница или приложение;
 - программа или команда DOS;
 - нужно ли вводить параметры при запуске приложения;
 - нужно ли ожидать нажатие на клавишу после окончания работы приложения для возврата в *Монитор*.

Являясь удобной средой для выбора и запуска программ (это доступно даже младшему школьнику), *Монитор* позволяет модифицировать свою оболочку, собирать в единый пакет разрозненные программы. Именно это свойство *Монитора* наряду с понятием “рабочего стола” позволило авторам поместить его изучение в программу первого семестра курса “Компьютерное конструирование”.

В качестве полезного побочного эффекта “конструкторские” способности *Монитора* помогут школьнику освоить файловую систему компьютера на практике.

Завершающим этапом по теме является конкурс пиктограмм, проводимый в форме дистанционного соревнования участников, в котором подводятся как командные, так и личные итоги.

Для участия в конкурсе команда должна выслать комплект пиктограмм для *Монитора*, разработанных детьми по теме “Школьные предметы и их разделы”. В учебнике приводится список примеров. Он демонстрирует, насколько широка тематика сюжетов.

Созданные в ходе конкурса пиктограммы могут предназначаться как для обозначения целых разделов (страниц *Монитора*), так и для вызова приложений с задачами по конкретным темам.

Набор пиктограмм должен сопровождаться текстовым файлом с описанием каждой из них по следующей схеме:

Заголовок файла:

- руководитель команды;
- регистрационный номер команды;
- e-mail команды.

Описание пиктограммы:

- название пиктограммы;
- назначение;
- имя файла;
- автор (авторы).

Пиктограммы (файлы с расширением CUP) и текстовый файл с их описанием следует послать в виде архива по сети в адрес конкурсного жюри в сроки, соответствующие календарному плану курса.

Работы, присылаемые на конкурс, оцениваются по некоторым критериям (см. ниже), которые коллективно обсуждаются. Это представляется тем более актуальным, что через несколько дней после завершения конкурса каждая команда получает пакет с работами команд-однокурсников, и руководителю команды вместе со своими учениками предстоит по тем же критериям проверить работы других команд.

Поэтому, проводя конкурс, учитель-руководитель должен рассказать об этих критериях школьникам. Тогда их стремление к высокому результату, вероятно, станет более осознанным. Можно даже попросить их сделать самооценку своих работ. В нынешнем конкурсе самооценка факультативна, однако в одном из последующих конкурсов самооценка включается в число заданий. При желании можно потренироваться в самооценке и в конкурсе пиктограмм.

Созданные детьми пиктограммы участвуют одновременно в двух конкурсах — личном и командном. При этом школьники не делают дополнительных заданий для участия в коллективном конкурсе. Формулы подсчета коллективной оценки оперируют исключительно теми оценками, которые детские работы получили в личном первенстве. Дополнительным “внешним” критерием в командной оценке учитывается только такой не детский труд, как учительское описание коллективной работы с представлением коллектива, его специфических особенностей, способностей отдельных учеников, их отношения к труду вообще и к теме “Монитор и пиктограммы” в частности.

Все работы школьников включаются в курсовой сборник, который слушатели могут получить с ftp-сервера. Лучшие работы формируют размещаемую на одной странице картинку, которая по завершении конкурса направляется участникам по курсовому списку рассылки.

При выставлении индивидуальной оценки экспертам рекомендуется оценивать:

- **практическую значимость** пиктограммы (если пиктограмма может быть рекомендована к практическому использованию в качестве метки какой-либо используемой в школе компьютерной программы, существующей, потенциально реализуемой или актуальной, то работа оценивается по этому критерию баллами 8—10 (10 — это максимальная оценка), для того чтобы претендовать на такую высокую оценку, применимость должна быть указана автором пиктограммы в ее описании; если описание отсутствует, но легко определяется практическим предназначением, пиктограмма получает оценку 5—7; если применимость пиктограммы не ясна или вызывает сомнения, пиктограмма получает оценку 2—4);

- **смысловую выразительность** (максимальный балл по этому критерию — 8); балл 6—8 получает пиктограмма, однозначно определяющая то явление, процесс или предмет, которое должна помечать пиктограмма; если соответствие пиктограммы и помечаемого объекта не однозначно, пиктограмма получает 4—5 баллов; если соответствие, по мнению эксперта, отсутствует или противоречит замыслу, описанному автором, результат оценивается баллом 1—3;
- **художественное (графическое) представление** пиктограммы; максимальное количество баллов — 8; в интервале от 1 до 8 эксперту предлагается оценить цветовую гамму рисунка-пиктограммы и ее композицию;
- **оригинальность**; самостоятельно придуманные и реализованные пиктограммы получают высший балл — 4; если пиктограмма близка по замыслу к одной из пиктограмм, используемых на панелях поставочного комплекта *Конструктора*, и несколько модифицирована, ее оценивают баллом 2—3; полностью скопированная пиктограмма оценивается баллом 1;
- наличие **описания**; максимальная оценка — 6 баллов; хорошее нестандартное описание — 4; краткое точное описание — 2 балла; неточное или ошибочное описание — 1 балл; отсутствие описания — 0 баллов. В описании надо указать фактические параметры пиктограммы — имя, название, автор (фамилия, имя, класс), — а также назначение пиктограммы (в каких предметах и темах, для обозначения каких объектов)

возможно ее использование. Обоснование выбора сюжета — наиболее красивая и весомая часть описания. Стихотворные описания вполне допустимы. Безусловно, описание, составленное детьми, ценится выше, чем учительское. Различие этих двух типов обычно сразу заметно, независимо от искусства маскировки, но, безусловно, 0 (отсутствие описания) — это ниже даже плохо составленного руководителем описания.

Таким образом, максимальное количество баллов, которое может получить пиктограмма:

Полезность	10
Выразительность	8
Художественность	8
Оригинальность	4
Документированность	6
Всего	36

Коллективная оценка команды формируется из индивидуальных оценок учащихся как средний балл участников команды.

Хотя роль руководителя команд формально не оценена даже в коллективном зачете, не надо убеждать читателя в значимости этой роли. Каждый руководитель должен сопровождать конкурсные работы описанием подготовки и организации конкурса (в свободной форме); подробной количественной, возрастной, психологической характеристике коллектива; впечатлениях детей и обо всем другом, что может представлять интерес для однокурсников.

Продолжение читайте в № 6/2002



<p>Гл. редактор С.Л. Островский Зам. гл. редактора А.И. Сенокосов Редакция: Е.В. Андреева Н.Л. Беленькая Л.Н. Картелишвили Н.П. Медведева Дизайн и верстка: Н.И. Пронская Корректоры: Е.Л. Володина, С.М. Подберезина</p>	<p>©ИНФОРМАТИКА 2002 выходит четыре раза в месяц При перепечатке ссылка на ИНФОРМАТИКУ обязательна, рукописи не возвращаются</p>	<p>Адрес редакции и издателя: 121165, Киевская, 24 тел. 249-48-96 Отдел рекламы тел. 249-98-70</p>	<p>Учредитель: ООО "Чистые пруды" Зарегистрировано в Министерстве РФ по делам печати. ПИ № 77-7230 от 12.04.2001. Отпечатано в ОИД "Медиа-Пресса", 125993, ГСП-3, Москва, А-40, ул. "Правды", 24. Тираж 7000 экз. Срок подписания в печать по графику 16.01.2001. Номер подписан 16.01.2001. Заказ № Цена свободная</p>
<p>ИНДЕКС ПОДПИСКИ для индивидуальных подписчиков 32291 комплекта изданий 32744</p>			
<p>Тел.: (095)249-31-38, 249-33-86. Факс (095)249-31-84</p>			
<p>Internet: inf@1september.ru WWW: http://www.1september.ru</p>			

**ИЗДАТЕЛЬСКИЙ
ДОМ
«ПЕРВОЕ
СЕНТЯБРЯ»,
ГЛАВНЫЙ
РЕДАКТОР —
А.СОЛОВЕЙЧИК**

Газеты ИЗДАТЕЛЬСКОГО ДОМА: **Первое сентября** — гл. ред. Е.Бирюкова, **Английский язык** — гл. ред. А.Громушкина, **Библиотека в школе** — гл. ред. О.Громова, **Биология** — гл. ред. Н.Иванова, **Воскресная школа** — гл. ред. монах Киприан (Яценко), **География** — гл. ред. О.Коротова, **Дошкольное образование** — гл. ред. М.Аромштам, **Здоровье детей** — гл. ред. А.Лекманов, **Информатика** — гл. ред. С.Островский, **Искусство** — гл. ред. Н.Исмаилова, **История** — гл. ред. А.Головатенко, **Литература** — гл. ред. Г.Красухин, **Математика** — гл. ред. И.Соловейчик, **Начальная школа** — гл. ред. М.Соловейчик, **Немецкий язык** — гл. ред. М.Бузоева, **Русский язык** — гл. ред. Л.Гончар, **Спорт в школе** — гл. ред. Н.Школьников, **Управление школой** — гл. ред. А.Адамский, **Физика** — гл. ред. Н.Козлова, **Французский язык** — гл. ред. Г.Чесновицкая, **Химия** — гл. ред. О.Блохина, **Чудесная газета** — гл. ред. М.Аромштам, **Школьный психолог** — гл. ред. М.Сартан.