

ИНФОРМАЦИОННАЯ КУЛЬТУРА

МОДУЛЬ

КЛАСС

10

МЕТОДИЧЕСКОЕ
ПОСОБИЕ



ПОД РЕДАКЦИЕЙ

А.Г.КУШНИРЕНКО, М.Г.ЭПИКТЕТОВА



МОСКВА
ИЗДАТЕЛЬСКИЙ ДОМ

Д Р О Ф А

1 9 9 5

УДК 373:002
ББК 74.261.6
И74

*Материалы книги разработаны и подготовлены
в объединении ИнфоМир*

Авторы:

А. Г. Кушниренко, А. Г. Леонов, М. Г. Эпиктетов,
В. В. Борисенко, М. А. Кузьменко,
Б. А. Назаров, С. В. Ханжин

Информационная культура: Модуль: Класс 10:
И74 Методическое пособие / А. Г. Кушниренко, А. Г.
Леонов, М. Г. Эпиктетов и др. — М.: Дрофа,
1995.—80 с.

ISBN 5—7107—0595—0

В книге представлены поурочные методические рекомендации для учителей, которые ведут уроки в десятом классе общеобразовательной школы по новому школьному курсу «Информационная культура».

Курс, имеющий в первую очередь образовательное, общеразвивающее значение, охватывает 11 учебных лет. Предусмотрено несколько «точек входа», в которых можно подключиться к изучению «Информационной культуры»; 9-й класс — одна из «точек входа». Темы, предлагаемые в 10-м классе, продолжают и развивают материал 9-го класса.

ББК 74.261.6

© «Дрофа», 1995

Глава 1. ПЕРЕДАЧА ИНФОРМАЦИИ

Материал этой главы рассчитан на 6 уроков и тесно примыкает к материалам модуля «Кодирование информации» (9 класс). Следует отметить, что при отсутствии в школе регулярного выхода в какую-нибудь компьютерную сеть материал придется преподавать «всухую», без практических примеров, и он может оказаться для учащихся очень трудным. Однако последующие главы не зависят от изложенного здесь материала, поэтому изучение главы 4 вполне можно значительно сократить.

Урок 1

Тема урока: Передача информации между людьми.

Программное обеспечение: Страница *Передача информации между людьми* из гипертекста к главе 1 (НХ1).

Цель урока: Ознакомление с современными методами передачи информации.

План урока:

1. Частоты колебаний.
2. Передача информации на большие расстояния.
3. Практическая работа с гипертекстом.

Домашнее задание.

1. Частоты колебаний.

Материал первого параграфа может быть кратко изложен учителем в начале урока. При обсуждении уместно изложить дополнительный материал, особый упор делая на понятия *ширина канала* и *мультиплексирование*.

Решение упражнения 1. Ответ: 10 МГц / 5 кГц = = 2000 разговоров одновременно.

Дополнительный материал

Кроме несущей частоты, важной характеристикой передачи является ширина полосы передаваемых частот — так называемая *ширина канала*. Так, при передаче разговора по телефону ширина канала составляет 3 килогерца, ширина телевизионного канала равна восьми мегагерцам. При передаче главная — несущая — частота модулируется частотами в пределах от нуля до ширины канала. Слово *модуляция* означает «наложение», т.е. передаваемый сигнал равен сумме несущего сигнала и модулирующих сигналов. При приеме несущая «вычитается» из общего сигнала. Таким образом, несущая частота не содержит никакой информации, информация передается с помощью модулирующего сигнала.

При фиксированной ширине канала несущая частота может быть любой, количество передаваемой в единицу времени информации, т.е. пропускная способность канала, при этом не меняется. Например, ширина любого из 12 каналов телевидения в нашей стране в точности равна восьми мегагерцам. Нижняя и верхняя границы первого канала телевидения равны 48,5 и 56,5 МГц, двенадцатого канала — 222 и 230 МГц; несущая частота в первом случае равна 49,75 МГц для изображения и 56,25 для звука, во втором случае — 223,25 и 229,75. Несмотря на значительную разницу частот, никакой разницы в качестве передачи по первому и двенадцатому каналам телевидения нет, поскольку ширина каналов в обоих случаях одинакова.

Независимость модулирующего сигнала и несущей частоты лежит в основе так называемого *мультиплексирования*, т.е. передачи по одному кабелю нескольких сигналов одновременно. Так, по телефонному кабелю передается одновременно 30 разго-

воров. Ширина каждого из 30 каналов равна трем килогерцам, промежуток между каналами — 2 кГц. Таким образом, несущие частоты отличаются на 5 кГц. Идея мультиплексирования иллюстрируется с помощью следующего примера. Пусть передаются несколько сигналов с помощью лучей света, каждому сигналу соответствует своя длина волны, т.е. свой цвет. Информация передается путем изменения интенсивности каждого сигнала. Передающее устройство собирает с помощью призмы несколько лучей разного цвета в один луч (оттенки которого постоянно меняются). Принимающее устройство также с помощью призмы разлагает этот луч на составляющие сигналы.

2. Передача информации на большие расстояния.

Телевидение вначале основывалось почти исключительно на передаче информации с помощью радиоволн. Напротив, телефонная сеть всегда использовала кабели, поскольку телефоны появились еще до изобретения радио. В настоящее время ситуация меняется на противоположную: практически во всех крупных городах США используется только кабельное телевидение (хотя и сохраняется телетрансляция); наоборот, телефоны становятся беспроводными (что, конечно, гораздо удобнее, поскольку их можно всюду носить с собой). Тенденция развития телефонной сети состоит также в переходе от аналоговой к цифровой передаче: так, в беспроводной сотовой телефонной сети используется цифровая передача.

Современные глобальные компьютерные сети ориентируются на оптоволоконную технологию: информация передается с помощью светового луча, генерируемого лазером, по гибкому оптическому кабелю из стекловолокна. Оптоволоконная технология пос-

тепленно вытесняет спутниковые системы, которые используются большей частью для пересечения океанов (хотя имеются и трансатлантические кабели).

Возможно, в будущем телевизионная, телефонная и компьютерная сети будут объединены в одну общую сеть, основанную на единой технологии, в частности на цифровой передаче информации.

3. Практическая работа с гипертекстом.

Учитель должен напомнить назначение клавиш **Enter** (выбор поля) и **Esc** (возврат к предыдущей странице) и запустить гипертекст к главе 1. Учащиеся должны поработать со страницей *Передача информации между людьми*, вспоминая назначения клавиш для работы с гипертекстом.

Домашнее задание.

Учитель может посоветовать ученикам ознакомиться по научно-популярной литературе с современными достижениями человечества в области передачи информации.

Урок 2

Тема урока: Двоичное кодирование (повторение).

Программное обеспечение: Страница *Двоичное кодирование* из гипертекста к главе 3 модуля 9 класса (НСЗ).

Цель урока: Повторение некоторых аспектов двоичного кодирования информации.

План урока:

1. Двоичное кодирование информации.
2. Практическая работа с гипертекстом.

Домашнее задание.

1. Двоичное кодирование информации.

Поскольку весь остаток четвертой главы подразумевает под *передачей информации* передачу дво-

ичных кодов, урок 2 следует посвятить повторению некоторых важных аспектов двоичного кодирования информации. Особое внимание следует уделить следующим моментам (подробнее см. методические рекомендации к главе 3 модуля 9 класса):

- с помощью двоичного кодирования любая информация может быть представлена в виде последовательности нулей и единиц;
- «правила игры» запрещают использование при кодировании каких-либо дополнительных символов (разделителей и т.п.).

2. Практическая работа с гипертекстом.

Учащиеся должны поработать со страницей *Двоичное кодирование* гипертекста НСЗ, решая упражнения. Учитель должен проконтролировать результаты работы учащихся.

Домашнее задание.

Дома ученикам рекомендуется освежить в памяти главу 3 из учебника для 9 класса, основной упор делая на двоичное кодирование.

Урок 3

Тема урока: Протоколы передачи.

Программное обеспечение: Страница *Протоколы передачи* из гипертекста к главе 1 (НХ1).

Цель урока: Ознакомление с понятием протокола.

План урока:

1. Среда передачи.
2. Понятие протокола.
3. Практическая работа с гипертекстом.

Домашнее задание.

1. Среда передачи.

Работа даже самых первых ЭВМ была бы невозможна без организации передачи данных между разными устройствами, входящими в состав самой ЭВМ. Так что можно считать, что сети появились вместе с появлением компьютеров.

Обязательной составляющей любой ЭВМ является *шина*, через которую подключаются и обмениваются данными различные устройства. При этом по шине передаются не только сами данные, но и адреса устройств или элементов памяти, которым эти данные предназначены. Передача происходит по четко установленным правилам, которые называются *интерфейсом*, или *протоколом*, работы шины. Наличие такого единого, четко описанного интерфейса позволяет независимо разрабатывать и подключать к ЭВМ самые различные устройства.

Сегментом называется участок кабеля, к которому возможно подключение компьютеров локальной сети без использования специальных устройств для расширения сети — повторителей, ретрансляторов, мостов, шлюзов и т.п. Обычно к одному сегменту подключается 10—20 компьютеров.

Максимальная длина сегмента оптоволоконного кабеля значительно превосходит максимальную длину сегмента электрического кабеля и измеряется сейчас десятками километров. (Для наиболее распространенных сетей типа Ethernet максимальная длина сегмента электрического кабеля составляет 200 метров для кабеля типа «тонкий Ethernet», 500 метров — для кабеля «толстый Ethernet» и 100 метров — для кабеля «витая пара».)

Что же касается пропускной способности оптоволоконного кабеля, то она поистине фантастическая. Сейчас стандартом является пропускная способность 100 мегабит в секунду (против 10 мегабит для

электрического кабеля типа Ethernet), но, несомненно, в ближайшее время она значительно возрастет. Кроме того, оптоволоконный кабель идеально защищен от помех и по цене не очень сильно превышает обычный электрический. Если же рассматривать стоимость передачи единицы информации, то оптоволоконная технология уже сейчас самая дешевая.

Кроме среды передачи, важной характеристикой сети является ее *топология* (или схема, конфигурация). Большие локальные сети (например, сеть внутри банка) строятся обычно по следующей схеме: имеется главный кабель — так называемая *магистраль* (backbone). Обычно это кабель типа «толстый Ethernet» или, что предпочтительнее, оптоволоконный кабель.

Подключение к магистрали осуществляется с помощью *трансиверов*, представляющих в случае кабеля «толстый Ethernet» коробку, которая надевается прямо на кабель; специальная иголка при этом протыкает кабель, обеспечивая таким образом подключение к нему. К трансиверам подключают специальные устройства — так называемые *хабы* (hub). Слово *hub* в переводе означает ступицу колеса, из которой торчат спицы. Хаб имеет 8—16 выходов для подключения компьютеров и один вход, через который он подключается к трансиверу на магистрали. Каждый компьютер подключается к хабу с помощью отдельного провода типа «витая пара» длиной до 100 метров.

Конечно, это только примерная схема, любая реальная сеть имеет свои особенности, а количество вариантов построения сложной сети неисчислимо.

2. Понятие протокола.

Основное понятие, которое ученики должны освоить на третьем уроке, — это понятие *протокола*.

Главная особенность компьютерных протоколов — их иерархическое описание.

Разработкой и стандартизацией протоколов передачи данных занимаются специальные международные организации, координирующие работу научных коллективов в разных странах. Разработка и соблюдение стандартов в области передачи данных важны, как ни в какой другой области. Протоколы передачи данных — это интернациональный язык, на котором общаются все компьютеры мира. Крупные компьютерные сети не знают национальных границ.

Разработка и стандартизация протоколов требует высокого научного и технического уровня. Принятие неудачных стандартов может сильно осложнить развитие технологии передачи данных.

Полезно будет обсудить упражнение 5 (*Ответ: при изменении скорости передачи данных достаточно изменить только протокол передачи байтов*).

3. Практическая работа с гипертекстом.

Учащиеся должны поработать со страницей *Протоколы передачи*.

Домашнее задание.

Придумать протокол передачи файлов (на основании протокола передачи байтов). При этом особое внимание надо уделить однозначности формулировки правил и независимости от протокола более низкого уровня.

Прочитать пункты 2.4—2.6.

Урок 4

Тема урока: Компьютерные сети.

Программное обеспечение: Страница *Компьютерные сети* гипертекста НХ1.

Цель урока: Формирование представления о компьютерных сетях.

План урока:

1. Обсуждение домашнего задания.
 2. Протоколы в компьютерных сетях.
 3. Виды компьютерных сетей.
 4. Практическая работа с гипертекстом.
- Домашнее задание.

1. Обсуждение домашнего задания.

Ученики должны понять, что основной вопрос, который надо решить при составлении протокола, — как отмечать конец файла. Можно для этой цели использовать какой-нибудь специальный символ, посылаемый в конце передачи файла (однако этот метод годится только для текстовых файлов), либо посылать в начале длину файла, а уже потом — его содержимое.

В реально используемых протоколах файл обычно делится на части. К каждой части добавляется ее длина и *контрольная сумма* (сумма значений посылаемых байтов по модулю 256). Это позволяет проконтролировать правильность приема информации. В случае возникновения ошибки (например, из-за помех в канале передачи) можно повторить передачу неверно принятой части файла.

2. Протоколы в компьютерных сетях.

Уровневая модель описания протоколов принята Международной организацией по стандартам — ISO (International Standard Organization). Международный стандарт «Взаимодействие открытых систем» (OSI — Open System Interconnection) выделяет семь уровней описания протоколов: физический, канальный, сетевой, транспортный, сессионный, уровень представления данных и прикладной (самый высокий). Конкретные протоколы должны описываться

в рамках этой схемы (правда, некоторые распространенные протоколы, существовавшие еще до модели OSI, не вполне в нее укладываются). Смысл этой модели состоит в том, что протокол более высокого уровня реализуется и описывается на базе протокола уровня на единицу меньше. При этом базовый протокол представляет собой «черный ящик»; верхний протокол ничего не должен знать о реализации базового протокола и не может обращаться к протоколам более низких уровней. Например, протокол сессионного уровня, то есть уровня взаимодействия между разными программами на одном или разных компьютерах, реализуется на базе протокола транспортного уровня, который осуществляет доставку пакетов (содержащих определенное количество информации и служебный «конверт») между различными адресами.

Такая модель позволяет изменить реализацию протокола нижнего уровня, не меняя протоколы верхних уровней. Например, можно заменить обычные сетевые платы Ethernet на всех компьютерах сети на более быстрые современные платы Fast Ethernet, обеспечивающие скорость 100 Мбит/сек (против 10 Мбит для Ethernet). При этом нужно будет заменить только драйвер протокола канального уровня; все остальное дорогостоящее сетевое программное обеспечение останется таким же, как и раньше.

Локальная сеть Ethernet устанавливает протоколы физического и канального уровней. В них описываются типы кабелей (коаксиальный или витая пара), формы и частоты электрических сигналов (несущая частота — 20 МГц, что соответствует скорости передачи 10 Мбит/сек), а также способы соединения компьютеров в сеть. В сети Ethernet все компьютеры под-

соединяются к кабелю вдоль одной линии — это так называемая топология шины. Порядок подсоединения разных компьютеров совершенно не важен. Информация передается в виде пакетов, размер пакета не может превышать 1518 байт и должен быть не меньше 64 байт. Служебная часть пакета содержит преамбулу, начальный ограничитель кадра, адрес получателя (6 байт), адрес отправителя (6 байт) и длину кадра. Пакет содержит данные и в конце — контрольную сумму. Ethernet — сеть широковещательного типа: при передаче пакета одним из компьютеров сети его могут принимать одновременно все остальные компьютеры.

Центральная часть любого протокола канального уровня — метод получения доступа к среде. Дело в том, что в каждый момент времени только один компьютер сети может передавать данные. Каждый компьютер, который имеет пакет для передачи, проверяет наличие несущей частоты в канале (наличие несущей означает, что в данный момент другой компьютер ведет передачу, т.е. канал занят). При пропадании несущей (освобождении канала) компьютер ждет некоторое время (9,6 микросекунды) и, если несущая за это время не появилась, начинает передавать пакет.

Неприятность состоит в том, что возможна ситуация, когда несколько компьютеров одновременно начнут передачу. Такая ситуация называется *коллизией*. При возникновении коллизии все компьютеры прекращают передачу, ждут случайное время и затем возобновляют попытку передать пакет в канал. Из-за того что случайное время ожидания различно для разных компьютеров, повторная коллизия менее вероятна. Всего компьютер может сделать не более 16 попыток передать пакет.

Над одним и тем же протоколом канального уровня (например, Ethernet) могут быть реализованы различные протоколы более высоких уровней. Например, наиболее распространенные локальные сети NetWare используют протокол транспортного уровня IPX; сети Unix — протокол IP. Протокол транспортного уровня описывает доставку пакетов между различными узлами так называемой интeрсети — объединения нескольких локальных сетей. Собственно, протокол транспортного уровня не содержит ничего, кроме формата адресов и формата пакетов. Реализация этого протокола должна обеспечивать доставку пакетов по назначению. Работа протокола транспортного уровня аналогична работе обычной почты.

Отметим, что транспортный уровень расположен выше сетевого. Это означает, что пакет может проходить через много различных разнородных сетей (объединенных в так называемую *интeрсеть* с помощью специальных компьютеров, которые называются *мостами* или *шлюзами*), прежде чем дойдет до адресата.

Сети LAN Manager, LANtastic, Windows for Work Groups построены над протоколом еще более высокого уровня — NETBIOS. Это протокол сессионного уровня, т.е. уровня взаимодействия между программами, работающими на разных компьютерах. Протокол NETBIOS обеспечивает установку виртуального (воображаемого) соединения — «сессии» — между программами и пересылку сообщений произвольной длины. Протокол сам разбивает длинное сообщение на последовательность пакетов со стороны передающей программы и вновь собирает это сообщение на стороне принимающей программы. При этом обеспечивается посылка подтверждений успешного приема пакетов и повторная передача пакета в случае обна-

ружения ошибки.

Сетевые устройства должны продвигать пакет ближе к месту назначения. В сетях со сложной топологией существуют специальные таблицы (алгоритмы) маршрутизации, которые позволяют по адресу назначения пакета определить следующий ближайший узел, на который его нужно передать.

3. Виды компьютерных сетей.

Материал пунктов 2.4–2.6 носит обзорный характер. Если есть возможность, его надо проиллюстрировать примерами из реальной практики (при наличии локальной сети в классе — рассмотреть, к какому типу она относится; при наличии выхода в РЕЛКОМ или другую почтовую сеть — разобрать формат записи адреса и структуру письма, и т.д.).

Дополнительный материал

Локальные и глобальные сети. В нашей стране наиболее популярна глобальная сеть РЕЛКОМ (от слов RELiable COMmunications — надежная связь). Эта сеть является продолжением сети Internet. Основные узлы сети РЕЛКОМ являются узлами сети Internet и подключены к ней с помощью современных каналов связи — спутниковых или оптоволоконных. Это компьютеры с операционной системой Unix, работающие круглосуточно. Индивидуальные пользователи сети РЕЛКОМ подключаются к этим узлам, используя телефонную сеть и модемы. Почта, которая приходит отдельным пользователям, накапливается на узлах сети РЕЛКОМ, пока пользователь не позвонит и не заберет ее.

Одноранговые сети и сети типа «клиент-сервер». Разделение на одноранговые сети и сети «клиент-сервер» достаточно условно: одноранговая сеть может использовать один или несколько серверов, в то

же время один сервер может быть клиентом другого. В одноранговой сети также всегда есть возможность указать, какие ресурсы будут разделяться между компьютерами сети, а какие будут доступны только на локальных компьютерах.

Наиболее распространенная сетевая операционная система типа «клиент-сервер» — это система NetWare фирмы Novell. На файловом сервере NetWare запускается своя операционная система, которая хранит данные о пользователях и осуществляет отдельный доступ к файлам (пользователь не может уничтожить не принадлежащий ему файл или даже узнать о том, что он существует, если только файл специально не открыт его владельцем). На файловом сервере можно запускать специальные программы — так называемые загружаемые модули (NLM — NetWare Loadable Module), которые могут обеспечивать выполнение других услуг, например, электронную почту (MHS-сервер) или работу с базами данных (SQL-сервер).

Клиентом NetWare может быть компьютер практически с любой операционной системой (MS DOS, Windows, Unix, MacOS и т.д.) — достаточно только загрузить программы, обеспечивающие доступ к серверу. После того как сервер опознает входное имя и проверит пароль, вы сможете «видеть» диски сервера как диски своего собственного компьютера — правда, только те файлы на них, которые принадлежат вам (либо открыты для общего использования).

Очень важно отметить, что «настоящая» сеть всегда предоставляет сервис более высокого уровня, чем просто пересылка файлов. (Более того, в самых распространенных сетях NetWare команда пересылки файлов между разными узлами сети просто отсутствует.) Сеть в качестве посредника между вашим

компьютером и ресурсами другого компьютера как бы «исчезает» (программисты употребляют термин «прозрачный доступ»). С точки зрения пользователя, на его компьютере появляется большое количество дополнительных дисков (которые на самом деле расположены на других компьютерах сети, но для пользователя они ничем не отличаются от обычных дисков). Опыт показывает, что очень часто начинающие пользователи даже не знают, что они работают в сети; и даже опытные пользователи часто не знают названия сетевого программного обеспечения, с которым они работают.

3. Практическая работа с гипертекстом.

Учащиеся должны поработать со страницей *Компьютерные сети*. Материал страницы гипертекста носит чисто информационный характер.

Домашнее задание.

Учитель советует ученикам выяснить, какие глобальные сети есть в данной местности.

Прочитать пункты 2.7—2.8.

Урок 5

Тема урока: Практическая работа с сетью.

Цель урока: Закрепление знаний о компьютерных сетях.

План урока:

1. Сравнение почтовой и компьютерной сети.
2. Мозаика.
3. Практическая работа с сетью.

Домашнее задание.

1. Сравнение почтовой и компьютерной сети.

Современные компьютерные сети — это так называемые *сети с коммутацией пакетов* (packet switching networks). Слово *коммутация* здесь не

слишком удачно (более правильно было бы говорить «сети с передачей пакетов» или «сети с обменом пакетами»), но оно стало общепринятым. Сети с коммутацией пакетов противопоставляются *сетям с коммутацией каналов* (circuit switching), которые в настоящее время почти не встречаются.

В сетях с коммутацией каналов перед передачей данных необходимо установить соединение между двумя участниками передачи — механическим или электронным путем. Пример сети с коммутацией каналов — обычная телефонная сеть. Перед началом разговора по телефону устанавливается соединение, которое действует на протяжении всего разговора. Такой способ взаимодействия плохо подходит для общения компьютеров (а точнее, для общения компьютерных программ: ведь на одном компьютере обычно работают одновременно несколько программ, и именно программы, а не компьютеры общаются друг с другом). Главные недостатки таких сетей:

- установка соединения занимает очень много времени по сравнению с быстродействием компьютерных программ;
- на все время взаимодействия канал занят и не может совместно использоваться другими программами.

Аналогия с телефоном: во время пауз в разговоре можно было бы передавать по тому же каналу другие разговоры, увеличивая реальную пропускную способность линии.

В противоположность сетям с коммутацией каналов, в сетях с коммутацией пакетов канал почти в любой момент времени открыт для всех узлов сети — ведь отдельный узел может занять сеть только на время передачи одного пакета, которое составля-

ет тысячные доли секунды. В реальных сетях утилизация канала (т.е. доля времени, когда канал занят) составляет не более 20–30%. Длительные процедуры установки соединения вообще отсутствуют (если, конечно, не рассматривать протоколы верхних уровней, в которых устанавливается логическое соединение между программами).

Любой узел сети может практически мгновенно отправить пакет в сеть; после этого продвижением пакета к адресату занимается уже сама сеть (компьютеры или сетевые устройства), а передающая программа может в это время выполнять любую другую работу. При получении пакета специальная программа — сетевой драйвер — определяет, какой из программ, работающих на данном компьютере, этот пакет адресован. (Отметим, что *всегда* в любой сетевой операционной системе на одном компьютере параллельно работают несколько программ.) Сетевой драйвер тем или иным способом сообщает нужной программе о том, что для нее пришел пакет, и передает ей принятый пакет. Таким образом, сетевой драйвер — это как бы почтальон. Адреса пакетов всегда включают не только адрес компьютера, но и адрес программы внутри данного компьютера (аналогично номеру квартиры в доме).

Решение упражнения 7. Ответ: примерно 1200.

2. Мозаика

В Internet существует много других видов сервиса, которые обеспечиваются соответствующими протоколами. Например, протокол telnet позволяет подключаться к удаленной машине и работать на ней, протокол ftp — переписывать файлы с одного компьютера на другой. Существует программа (и протокол) gopher, которая организует работу с меню. Мозаика объединила большинство существовавших

ранее видов сервиса, а работа с помощью Мозаики намного удобнее. Так, программой ftp теперь уже почти не пользуются.

3. Практическая работа с сетью.

Урок следует посвятить практическому знакомству с возможностями какой-нибудь компьютерной сети. Это может быть работа в классе на имеющемся оборудовании (минимальное требование — наличие модема) либо экскурсия к коллегам.

Домашнее задание.

Домашнее задание не предусмотрено, однако учитель может посоветовать познакомиться с практической работой модема на работе у родителей или у друзей.

Урок 6

Тема урока: Что такое модем.

Программное обеспечение: Страница *Модем* гипертекста НХ1.

Цель урока: Формирование представления о том, как работает модем.

План урока:

1. Обсуждение принципов устройства модема.
2. Практическая работа с гипертекстом.

Домашнее задание.

1. Обсуждение принципов устройства модема.

Подробно разбирать материал § 3 имеет смысл только при наличии в школе модема и доступа к какой-нибудь сети или BBS (либо если есть договоренность о сотрудничестве с другой школой, также оснащенной модемом). В противном случае достаточно кратко рассмотреть начало § 3 и пункты 3.1—3.2.

Следует отметить опасность, которую несет с собой модем. Сотня-другая модемов, медленно перекачивающих огромные объемы информации, способны блокировать работу телефонной станции. Эта опасность не была столь явной, пока персональных компьютеров было немного, но теперь она весьма реальна.

Чтобы избавиться от недостатков существующих телефонных линий, был разработан новый стандарт телефонных сетей — ISDN, то есть интегрированные сервисные цифровые сети (Integrated Service Digital Networks). ISDN позволяют передавать как обычный звук, так и компьютерные данные в цифровом виде; скорость передачи равна 64 000 бит в секунду. Во время разговора звук преобразуется в цифровую форму и в таком виде передается по линии — то есть происходит преобразование, обратное тому, которое выполняет модем. К сожалению, ISDN были встречены весьма прохладно. Для развития ISDN требуются крупные первоначальные вложения, и большинство пользователей не идут на них, пока есть возможность работать с помощью обычного модема.

Возможно, другая причина сдержанного отношения к ISDN состоит в том, что скорость передачи, которую обеспечивают ISDN, тоже недостаточна для современных компьютеров. Она всего в три с небольшим раза больше скорости работы высококачественного модема (хотя, конечно, надежность ISDN неизмеримо выше). В настоящее время разработан новый стандарт на широкополосные ISDN — так называемые BSDN (от слова *Broadband*), которые обеспечивают высокую скорость передачи.

2. Практическая работа с гипертекстом.

Учащиеся должны поработать со страницей *Модем*, решая упражнения из гипертекста. Нет необ-

ходимости вникать в детали работы приведенных в учебнике алгоритмов (и тем более заучивать команды модема), достаточно разобрать их общую структуру.

Домашнее задание.

Прочитать введение к главе 2, а также введение и пункты 4.1–4.3 из § 4.

Глава 2. ИНФОРМАЦИОННЫЕ МОДЕЛИ

Эта глава в модулях 9–10 — ключевая. Ее материал опирается на тему «Повторим программирование» 9 класса, углубляет материал двух тем 9 класса по кодированию информации и готовит учащихся к теме «Система ПланиМир». Основной навык, который должны получить школьники после прохождения главы, — это умение в простейших ситуациях, в зависимости от круга решаемых задач, предложить несколько вариантов информационной модели.

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

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

Подобные задачи, в которых вначале нужно придумать «информационную» модель жизненной ситуации, а уж только потом составить программу получения ответа по этой модели, рассматриваются в данной главе. Задачи на придумывание информационных моделей, на наш взгляд, важнее задач на составление алгоритмов. К счастью, таких задач можно придумать много, и для большинства учеников

эти задачи интереснее и доступнее задач на составление алгоритмов.

Задачи на составление информационных моделей привлекательны тем, что очень просты технически и допускают, как правило, несколько равноприемлемых решений. Для каждой задачи по обработке информации, содержащейся в модели, решение может обсуждаться на двух уровнях. На первом уровне проводятся неформальные обсуждения того, разрешима ли задача вообще, достаточно ли в модели информации для решения задачи. Если выясняется, что информации недостаточно, то возникает необходимость в модификации информационной модели, а вовсе не в составлении алгоритма. Если же оказывается, что в принципе информации достаточно, то может идти речь о переходе к более строгому второму уровню, к составлению формального алгоритма на школьном алгоритмическом языке.

Важные рекомендации:

- а) при обсуждении каждой информационной модели желательно хотя бы одну задачу по обработке информации довести до второго уровня — составления алгоритма;
- б) вместе с тем не следует увлекаться доведением каждой задачи до решения на втором уровне. Гораздо полезнее обсудить для данной модели несколько задач по обработке информации неформально, чем потратить все время на трудное программирование интуитивно ясного алгоритма.

Выбор материала, на котором построено изложение в учебнике, один из многих возможных. С равным успехом тему можно проходить и на других примерах. Точно так же и задачи, приведенные в учебнике, без ущерба могут быть заменены други-

ми, предложенными учителем или школьниками. Единственное исключение — материал параграфа 18 «Информационные модели геометрической информации». Этот материал нужен для поддержки главы 7 учебника и программного обеспечения к этой главе и должен быть пройден достаточно полно.

Материал главы 2 рассчитан на 10 уроков, с 7-го по 16-й. Далее приводится один из многих возможных вариантов поурочного планирования.

Урок 7

Тема урока: Понятие информационной модели.

Программное обеспечение: Страница *Модель кинозала* из гипертекста к главе 2 (НХ2).

Цель урока: Формирование представления о понятии информационной модели.

План урока:

1. Обсуждение определения информационной модели.
2. Разбор примера информационной модели.
3. Внесение в модель информации о ценах.
Домашнее задание.

1. Обсуждение определения информационной модели.

На определении информационной модели не следует долго задерживаться. Это определение будет усвоено учениками позже, после рассмотрения примеров. Три особенности понятия *информационная модель* также будут поняты позже, но в момент введения определения их следует перечислить.

1) Определение информационной модели в значительной мере привязано к используемому языку программирования, так как модель строится средствами используемого языка. Чем богаче язык, тем легче составить информационную модель.

2) Хотя модель и привязана к языку программирования, но в модель входят и правила, связывающие заключенную в модели информацию с реальными процессами, объектами или явлениями. Эти связи по необходимости описываются неформально, на русском языке (быть может, с вкраплениями математической и программистской символики). Так что задача составления информационной модели не может иметь чисто формальное решение.

3) Модель строится не сама по себе, а с учетом круга решаемых задач. Модель должна быть *достаточной* для выбранного круга задач, должна содержать достаточно информации. Как правило, для данного круга задач информационную модель можно строить многими разными способами.

Посвятив несколько минут урока 7 определению, можно перейти к разбору простейшего примера информационной модели (§ 4).

2. Разбор примера информационной модели.

Хотя для краткости в учебнике всюду говорится про информационную модель кинозала, правильнее было бы сказать, что строится *модель процесса продажи билетов* в кинозал на один-единственный сеанс. Разобрав материал пункта 4.1, следует подчеркнуть такое обстоятельство.

Автор информационной модели всегда имеет как **права**, так и **обязанности**. Они тесно связаны между собой. Автор имеет **право** выбрать информационную модель по своему вкусу. Но после выбора модели автор **обязан** реализовать алгоритмы для решения того круга задач, для которого составлялась модель. При составлении алгоритма автору приходится работать с составленной им моделью, и если она неудачна, то ему некого винить, кроме самого себя.

На первых порах может вызвать затруднения двойной

цикл для перебора всех элементов прямоугольной таблицы. В таком случае полезно рассмотреть пример кинозала с 2 рядами по 3 места в каждом.

3. Внесение в модель информации о ценах.

Завершается урок 7 разбором пунктов 4.2 и 4.3.

Домашнее задание.

Составить алгоритм подсчета выручки по таблицам $a[1:560]$ и $цена[1:28]$. Прочитать пункты 4.4—4.6.

Урок 8

Тема урока: Простейшая модель кинозала.

Программное обеспечение: Страница *Модель кинозала* гипертекста НХ2.

Цель урока: Ознакомление с простейшей моделью кинозала и ее расширениями.

План урока:

1. Обсуждение домашнего задания.
 2. Разбор алгоритмов подсчета числа проданных билетов и выручки.
 3. Какую информацию учитывает модель.
 4. Расширение модели информацией о проходах.
- Домашнее задание.

1. Обсуждение домашнего задания.

Одно из возможных решений заданной на дом задачи таково:

```
s:=0
нц для j от 1 до 28
  нц для i от 1 до 20
    если a[i+(j-1)*20]=1
      то s:=s+цена[j]
    все
  кц
кц
```

Оно отличается от приведенного в пункте 4.2 только тем, что вместо простого и понятного $a[i, j]$ приходится писать $a[i+(j-1)*20]$.

2. Разбор алгоритмов подсчета числа проданных билетов и выручки.

Далее можно переходить к разбору алгоритмов пункта 4.4. После того как ученики разберут алгоритмы, полезно переключить их внимание на проблему связи модели с реальной действительностью.

Сделать это можно в процессе обсуждения такой гипотетической ситуации. В кинозале было 29 рядов по 22 места в каждом. В процессе проверки кинозала пожарной инспекцией было обнаружено, что проходы вдоль стен и перед экраном слишком узкие, и дирекции кинотеатра пришлось демонтировать целиком первый ряд, а также первое и последнее кресло в каждом ряду. После этого в зале осталось 28 рядов по 20 мест в каждом, но в целях экономии средств таблички с указанием номеров мест и рядов не переставлялись.

Формально, рассмотренная в пункте 4.1 модель годится и для этого случая. Просто при построении модели считаем номером ряда его порядковый номер, отсчитываемый от экрана, а номером кресла считаем его порядковый номер в ряду, отсчитываемый слева направо. При таком формальном подходе не учитываются таблички с номерами, реально привинченными к креслам. И чтобы перейти от информации внутри модели к информации в реальном мире, каждый раз придется различать *внутренние, модельные* номера ряда и места в ряду и *реальные* номера в кинозале, материализованные в табличках на креслах. И эти правила перевода внутренней информации во внешнюю придется включить в модель.

Чтобы этого избежать и упростить модель за счет

совпадения модельных и реальных номеров, в такой ситуации лучше использовать в модели таблицу

целтаб $a[2:21, 2:29]$

Такое изменение модели потребует переделки алгоритмов, но сложнее они не станут.

3. Какую информацию учитывает модель.

Далее следует разобрать материал пункта 4.5. Интуитивно алгоритм поиска пары соседних свободных мест очевиден, хотя и не так легко программируется. Но основное назначение этого пункта не в этом программировании, а во введении ситуации, когда уже построенная модель оказывается недостаточной для решения задачи и модель приходится расширять. Вот это-то расширение модели и есть основная задача урока 8.

4. Расширение модели информацией о проходах.

В заключение урока 8 разбирается материал пункта 4.6. Перед разбором алгоритма поиска свободных мест следует напомнить ученикам, что символами $\langle \rangle$ в алгоритмическом языке обозначается операция сравнения \neq . Если этот алгоритм вызовет затруднения, можно ограничиться его разбором на неформальном уровне.

Чтобы проконтролировать учеников, можно предложить им такую задачу. Первые 6 рядов кинозала разделены на две части — левую и правую, а остальные ряды разделены на 3 части, как описано в учебнике. Надо расширить информационную модель так, чтобы можно было искать два соседних свободных места.

Домашнее задание.

Прочитать пункт 4.7, просмотреть 4.8.

Урок 9

Тема урока: Конкурирующие расширения модели.

Программное обеспечение: Страница *Модель кинозала* гипертекста НХ2.

Цель урока: Формирование представления о возможности различных способов построения модели.

План урока:

1. Рассмотрение различных расширений информационной модели кинозала.

2. Окончательная модель кинозала (начало).

Домашнее задание.

Основная задача урока: показать, что модель можно строить многими разными способами. Тут нужно напомнить о правах и обязанностях автора модели: он может как угодно выбрать модель, но должен реализовать в этой модели те алгоритмы обработки информации, для которых модель предназначалась.

1. Рассмотрение различных расширений модели кинозала.

Урок начинается с обсуждения чисто житейской ситуации: улучшение обслуживания пенсионеров в кинотеатре. После разбора первого способа расширения модели можно обсудить его недостатки. Далее следует перейти ко второму способу, обсудить изменения алгоритмов и перейти к проблеме выдачи бесплатных билетов. То, что выдачу таких билетов невозможно учесть, является убийственным недостатком новой модели и мотивирует введение еще одной модели.

2. Окончательная модель кинозала (начало обсуждения).

Начать рассмотрение окончательной модели кинозала можно в конце урока 9. Особое внимание на-

до обратить на то, что в этой модели все числовые параметры модели заданы при помощи величин. Это позволяет легко изменить их, не исправляя алгоритмы работы с моделью.

Домашнее задание.

Прочитать пункт 4.8.

Урок 10

Тема урока: Окончательная модель кинозала.

Программное обеспечение: Страница *Модель кинозала* гипертекста НХ2.

Цель урока: Ознакомление с окончательной моделью кинозала.

План урока:

1. Окончательная модель кинозала (завершение обсуждения).

2. Решение упражнений.

Домашнее задание.

1. Окончательная модель кинозала (завершение обсуждения).

В начале урока завершается обсуждение модели кинозала из пункта 4.8.

2. Решение упражнений.

Остаток урока следует посвятить решению и обсуждению упражнений 8—12. Учитель сам должен решить, какие из предложенных пяти упражнений нужно решить в классе, а какие достаточно кратко обсудить и задать на дом.

Домашнее задание.

Упражнения к § 4 (по выбору учителя). Просмотреть § 5.

Урок 11

Тема урока: Информационная модель транспортной сети.

Программное обеспечение: Страница *Модель транспортной сети* гипертекста НХ2.

Цель урока: Знакомство с моделью, в которой простые по форме вопросы требуют для ответа сложных алгоритмов.

План урока:

1. Обсуждение домашнего задания.
 2. Разбор модели транспортной сети.
 3. Решение упражнений.
- Домашнее задание.

1. Обсуждение домашнего задания.

Урок следует начать с обсуждения домашнего задания. Как уже отмечалось, в задачах на составление информационных моделей решение, как правило, не единственное (хотя часто задача имеет «естественное» решение, которое проще всех остальных).

Решение упражнений:

9. Проще всего решить эту задачу, если ввести *наценку за места в центре* и считать, что цена билета складывается из цены за ряд (которая не зависит от номера места) и цены за место (которая, напротив, зависит от номера ряда). Однако это можно сделать не всегда — автор информационной модели не вправе изменять установленных цен на билеты. Поэтому в общем случае придется таблицу цен сделать двумерной:

`цел таб цена[1:imax, 1:jmax]`

где `цена[i, j]` — это цена билета на *i*-е место в *j*-м ряду. Заполнять такую таблицу, разумеется, сложнее, но алгоритм подсчета выручки в новой модели

почти ничем не отличается от алгоритма подсчета выручки в модели М1.

10. Как и в упражнении 9, имеется простое решение для случая, когда цена билетов на разные сеансы отличается на постоянную величину (или в постоянное число раз).

2. Разбор модели транспортной сети.

Остаток урока должен быть посвящен разбору модели, алгоритмов параграфа 5 и решению задач к этому параграфу. В этой модели самые естественные вопросы (например, каково минимальное количество посадок при полете из одного города в другой) могут потребовать довольно сложных алгоритмов. Необходимо, чтобы после работы с моделью продажи билетов, где все алгоритмы были очень просты, школьники столкнулись с ситуацией, когда простые по форме вопросы требуют для ответа сложных алгоритмов.

3. Решение упражнений.

Как и на предыдущем уроке, учитель должен сам выбрать упражнения из числа приведенных в § 5. Упражнения 13 и 14 довольно сложны, и будет вполне достаточно, если ученики смогут объяснить основную идею решения, не затрагивая технических подробностей.

Ниже приведены решения для всех упражнений. Рекомендуется на этом или на следующем уроке, после обсуждения упражнений, организовать просмотр готовых решений. Имена файлов с правильными решениями приведены ниже, перед текстом каждого алгоритма. Эти файлы можно загрузить в систему КуМир командой `Ctrl Alt F3` (либо выбрав пункт *Добавить фрагмент* из меню).

13а. Файл: 2TN1(A).E

алг список (арг цел i1,i2)нач цел iнц для i от 1 до nесли i <> i1 и прям [i1,i]="да" и i <> i2 ипрям [i,i2]="да"то вывод назв[i1], "-", назв[i], "-", назв[i2]+", "вывод "длина =", расст[i1,i]+расст[i,i2], нсвсекцкон

13б. Файл: 2TN1(B).E

алг список (арг цел i1,i2)нач цел i, jцелтаб m[1:n], цел n, иминвещтаб p[1:n], вещ рмин

n:=0

нц для i от 1 до nесли i <> i1 и прям [i1,i]="да" и i <> i2 ипрям [i,i2]="да"то n:=n+1

m[n]:=i

p[n]:=расст[i1,i]+расст[i,i2]

всекцнц для j от 1 до n

рмин:=1.E+10

нц для i от 1 до nесли p[i] < рмин то имин:=i; рмин:=p[i] всекц

p[имин]:=1.E+10

вывод назв[i1], "-", назв[m[имин]], "-", назв[i2]+", "вывод "длина =", рмин, нскцкон

13в. Файл: 2TN1(C).E

алг маршрут (арг цел i1,i2)нач цел i, jцелтаб m[1:n], цел n, иминвещтаб p[1:n], вещ рмин

n:=0

нц для i от 1 до nесли i <> i1 и прям [i1,i]="да" и i <> i2 ипрям [i,i2]="да"то n:=n+1

m[n]:=i

p[n]:=расст[i1,i]+расст[i,i2]

всекц

рмин:=1.E+10

нц для i от 1 до nесли p[i] < рмин то имин:=i; рмин:=p[i] всекцвывод назв[i1], "-", назв[m[имин]], "-", назв[i2]+", "вывод "длина =", рмин, нскон

13г. Файл: 2TN1(D).E

алг маршрут (арг цел i1,i2)нач цел i, j, лог нашли

нашли:=нет

нц для i от 1 до nнц для j от 1 до nесли i1 <> i и прям [i1,i]="да" и i <> j ипрям [i,j]="да" и j <> i2 и прям [j,i2]="да"то нашли:=давывод "Найден маршрут", назв[i1], "-", назв[i]вывод "-", назв[j], "-", назв[i2]всекц при нашликц при нашлиесли не нашли то вывод "Маршрут не найден" всекон

14. Файл: 2TN1(A).E

```
алг маршрут (арг цел i1,i2,м,целтаб р[1:м])
нач цел i,лог нашли
  нашли:=нет
  нц для i от 1 до м
    если прям[р[i],i2]="да"
      то нашли:=да
      вывод "Из",назв[i1],"в",назв[i2]
      вывод "можно добраться через",назв[р[i]]
    все
  кц_при нашли
  если не нашли то вывод "Маршрут не найден" все
кон
```

Домашнее задание.

Упражнения к § 5 (по выбору учителя). Просмотреть § 6.

Урок 12

Тема урока: Модели геометрической информации.

Программное обеспечение: Страница *Модели геометрической информации* гипертекста НХ2.

Цель урока: Формирование представления о способах построения информационных моделей для геометрической информации.

План урока:

1. Обсуждение домашнего задания.
2. Обсуждение геометрических моделей.
3. Перевод с геометрического языка на алгоритмический и обратно.
4. Построение моделей геометрических объектов в пространстве.

Домашнее задание.

1. Обсуждение домашнего задания.

Урок следует начать с разбора заданных на дом задач (см. пункт 3 предыдущего урока).

2. Обсуждение геометрических моделей.

В начале обсуждения информационных моделей геометрической информации необходимо бегло разобрать модели М3—М11. После этого полезно вернуться к модели М4 и придумать еще несколько моделей для задания окружности. Вот возможные варианты.

Модель М4'.

вещ a,b | координаты центра
вещ x,y | координаты точки на окружности

Модель М4''.

вещ x1,y1 | координаты 1-й точки на окружности
вещ x2,y2 | координаты 2-й точки на окружности
вещ x3,y3 | координаты 3-й точки на окружности

Модель М4'''.

вещ a,b | координаты центра окружности
вещ x1,y1 | координаты начала отрезка
вещ x2,y2 | координаты конца отрезка
(радиус окружности равен длине отрезка)

Важно отметить, что из этих моделей нельзя выбрать «наилучшую»: хотя модель М4 самая простая, она может оказаться не самой удобной для решения некоторых задач.

3. Перевод с геометрического языка на алгоритмический и обратно.

Далее следует заняться переводом с геометрического языка на алгоритмический и обратно. Наряду с приведенными в учебнике примерами и упражнениями можно использовать любые другие задачи и понятия из курса геометрии.

4. Построение моделей геометрических объектов в пространстве.

Остаток урока следует посвятить решению упражнения 21 из § 6.

Домашнее задание.

Прочитать § 7.

Урок 13

Тема урока: Модель обстановки на поле Робота.

Программное обеспечение: Страница *Модель обстановки на поле Робота* гипертекста НХ2.

Цель урока: Ознакомление с информационной моделью обстановки на поле Робота.

План урока:

1. Знакомство с Роботом (повторение).
2. Разбор информационной модели.
3. Решение упражнения 22.

Домашнее задание.

1. Знакомство с Роботом (повторение).

Школьники имели опыт работы с Роботом в 9 классе. Тем не менее полезно в начале урока потратить 10 минут для работы с гипертекстом 9 класса «Знакомство с Роботом».

2. Разбор информационной модели.

Далее можно переходить к разбору информационной модели по учебнику. Основная задача школьников на этом уроке — научиться связывать интуитивно ясные действия Робота и информацию на его поле с величинами модели.

3. Решение упражнения 22.

Из модели можно удалить, например, таблицы выше и правее. Проверка условий «сверху стена» и «справа стена» при этом ненамного усложнится.

Домашнее задание.

Прочитать § 8.

Урок 14

Тема урока: Модель рисунка на поле Чертежника.

Программное обеспечение: Страница *Модель рисунка на поле Чертежника* гипертекста НХ2.

Цель урока: Ознакомление с информационной моделью рисунка на поле Чертежника.

План урока:

1. Знакомство с Чертежником (повторение).
2. Разбор информационных моделей рисунка.
3. Решение упражнений.

Домашнее задание.

1. Знакомство с Чертежником (повторение).

В начале урока полезно потратить 10—15 минут на работу с гипертекстом «Знакомство с исполнителем Чертежник».

2. Разбор информационных моделей рисунка.

Далее можно перейти к информационным моделям M15 и M16. У доски полезно разобрать решение упражнения 24, несмотря на некоторую громоздкость решения. Наконец, можно обсудить еще один способ задания информационной модели рисунка на поле Чертежника: в виде программы на школьном алгоритмическом языке, управляющей Чертежником.

Такой способ полезно упомянуть, поскольку в главе 4 будет обсуждаться подобный способ задания чертежа в системе ПланиМир.

3. Решение упражнений.

Урок следует завершить решением упражнений 23 и 25, приведенных в конце параграфа.

25. В приведенном ниже решении таблица a15 берется из модели M15, а таблица a16 — из модели M16:

```
вещтаб a16[1:4000]
цел n | число отрезков,
      |  $0 \leq n \leq 1000$ ;
вещтаб a15[1:4000] | координаты отрезков.
алг Перевод из M16 в M15
нач цел i, k
i:=1 | начинаем просматривать a16 сначала
n:=0 | таблица a15 вначале пуста
k:=a16[i]; i:=i+1
нц пока k>0
  a15[4*n+1]:=a16[i]
  a15[4*n+2]:=a16[i+1]; i:=i+2
  нц k-1 раз
    a15[4*n+3]:=a16[i]
    a15[4*n+4]:=a16[i+1]; n:=n+1
    a15[4*n+1]:=a16[i]
    a15[4*n+2]:=a16[i+1]; i:=i+2
  кц
  a15[4*n+3]:=a16[i]
  a15[4*n+4]:=a16[i+1]; n:=n+1; i:=i+2
  k:=a16[i]; i:=i+1
кц кон
```

Домашнее задание.
Просмотреть § 9.

Уроки 15—16

Тема уроков: Решение задач на построение информационных моделей.

Программное обеспечение: Страница *Задачи на построение моделей* гипертекста НХ2.

Цели урока: Закрепление навыков составления информационных моделей.

На этих уроках можно использовать как задачи из § 9, так и собственные задачи. Уроки можно провести в форме конференции, где школьники делают заранее подготовленные доклады по задачам.

Домашнее задание.

Прочитать начало § 10 (пункты 10.1 и 10.2).
Просмотреть пункт 10.3.

Глава 3. КАК УСТРОЕНЫ КОМПЬЮТЕРНЫЕ ПРОГРАММЫ

Урок 17

Тема урока: «Записная книжка». Пример базы данных.

Программное обеспечение: Страница «Записная книжка» из гипертекста к главе 3 (НХ3).

Цель урока: Формирование представления о принципах устройства СУБД с точки зрения ее программирования.

План урока:

1. Повторение методов работы с базами данных.
2. Обсуждение принципов устройства СУБД.
3. Знакомство с исполнителем DB.

Домашнее задание.

В рамках курса «Информационная культура» ученики уже познакомились с базами данных (модуль 8 класса). Однако, в 8 классе базы данных рассматривались с «пользовательской» точки зрения, а теперь показывается другая точка зрения — «программистская». На примере несложной программы «Записная книжка» демонстрируются некоторые принципы создания прикладных баз данных.

1. Повторение методов работы с базами данных.

Начало параграфа (пункты 10.1—10.2) посвящено повторению основных свойств баз данных и систем управления базами данных (СУБД). Урок рекомендуется начать с практической работы по внесению данных в базу «Записная книжка».

На уроке можно также использовать материалы и программное обеспечение модуля 8 класса.

2. Обсуждение принципов устройства СУБД.

Далее нужно переходить к обсуждению принципов устройства СУБД. В процессе этого обсуждения надо подвести учеников к мысли о необходимости написания алгоритмов для обработки данных (в первую очередь это вытекает из требований сохранения целостности данных и их защиты). После этого можно переходить к рассмотрению основных операций, которые выполняют СУБД при работе с таблицами.

Реально используемые производственные СУБД (Fox-Pro, Clarion, Paradox, Oracle и другие), как правило, имеют свой собственный язык для записи алгоритмов работы с данными. В данном курсе не ставится задача подготовки профессиональных программистов, поэтому для демонстрации принципов устройства СУБД используется уже известный ученикам школьный алгоритмический язык.

3. Знакомство с исполнителем DB.

В конце урока надо начать знакомство с исполнителем DB. При обсуждении основное внимание следует уделять тому, как решаются основные задачи СУБД в этом исполнителе.

Домашнее задание.

Прочитать пункты 10.3 и 10.4.

Урок 18

Тема урока: «Записная книжка» (продолжение).

Программное обеспечение: Страница «Записная книжка» гипертекста НХЗ.

Цель урока: Формирование представления об устройстве программ работы с базами данных.

План урока:

1. Обсуждение системы команд исполнителя DB.
2. Практическое знакомство с программой «Записная книжка».
3. Анализ текста программы «Записная книжка».

Домашнее задание.

1. Обсуждение системы команд исполнителя DB.

Материал пункта 10.3, описывающий систему команд учебного исполнителя DB, не предназначен для заучивания. Этот материал фактически будет использован всего один раз, его достаточно кратко обсудить и потом использовать при необходимости как справочник.

В книге для ученика никак не затронут вопрос удаления таблиц (для решения поставленной задачи это все равно не потребуется). В исполнителе DB нет команды удаления таблицы, таблица автоматически удаляется, если в процессе работы из нее будут удалены все записи.

2. Практическое знакомство с программой «Записная книжка».

Знакомство с программой «Записная книжка» следует начать с практической работы по внесению информации в базу данных и поиску информации в базе. В процессе работы у учеников наверняка возникнет вопрос: «Как выполнить поиск только по одному полю (например, по имени)?» В зависимости от

уровня подготовки класса учитель может либо сразу дать ответ («Просто не надо заполнять все остальные поля»), либо дать ученикам догадаться самим (подсказка: надо рассмотреть команду начать поиск, в ней пустое значение строки обозначает, что значение может быть любым).

3. Анализ текста программы «Записная книжка».

При анализе текста программы достаточно разобрать подробно одну или две альтернативы в каждом выборе. Следует обратить внимание учеников на особенность программ работы с базами данных: они обычно состоят из большого числа достаточно независимых фрагментов, каждый из которых сам по себе довольно прост.

Решение упражнения 37. Программа просмотра данных о людях с днем рождения в определенную дату может выглядеть, например, следующим образом:

алг день рождения (арг лит дата)

нач цел к, лог усп, лит таб В[1:5], С[1:5]

```
В[1]:= " |
В[2]:= " | пустые строки означают, что значение
В[3]:= " | поля может быть любым
В[5]:= " |
В[4]:= дата
начать работу ("NOTEBOOK.DB", 5)
усп := начать поиск(В, С)
к := 0
```

нц пока усп и к <> 3

вывод данных(С)

вывод '1 - следующий, 2 - предыдущий'

вывод '3 - выйти ?>'

ввод к

вывод нс

выбор

при к = 1:

усп := следующий(С)

при к = 2:

усп := предыдущий(С)

все

кц

кон

В этой программе используется вспомогательный алгоритм вывод данных из программы «Записная книжка».

Домашнее задание.

Прочитать пункты 11.1–11.3 из § 11.

Урок 19

Тема урока: Графический редактор (начало).

Программное обеспечение: Страница *Графический редактор* гипертекста НХЗ.

Цели урока: Ознакомление с графическим редактором, формирование представления о принципе построения программы с использованием готовых блоков.

План урока:

1. Зачем нужен графический редактор?
2. Исполнитель ГраТекс.
3. Режим записи «исключающее ИЛИ».
4. Исполнитель Курсор.

Домашнее задание.

1. Зачем нужен графический редактор?

Урок следует начать с обсуждения пункта 11.1 и практической работы в графическом редакторе.

2. Исполнитель ГраТекс.

На этом уроке необходимо обратить особое внимание на принцип построения программы с использованием готовых блоков (в данном случае — исполнителей ГраТекс и Курсор). В наше время, когда

программные проекты выросли до невообразимых размеров, это становится уже необходимостью. Особенно это заметно в объектно-ориентированных системах, когда программы пишутся с использованием многочисленных библиотек классов (КуМир является отчасти объектно-ориентированной средой, в которой роль классов играют исполнители).

Графический редактор, описываемый в уроках 19—21, обращается с физическим оборудованием (видео, клавиатура) не напрямую, а уже используя готовые блоки. Таким образом, главное в редакторе — логические конструкции, общая логика управлением программой, а вся работа с «железом» (hardware) реализована отдельно и может быть исправлена или изменена независимо от основной программы.

3. Режим записи «исключающее ИЛИ».

Наиболее сложным понятием в исполнителе ГраТекс является понятие режима записи, и его стоит обсудить отдельно.

Режимы записи «логическое И» и «логическое ИЛИ» в монохромном режиме совершенно бесполезны (при их использовании на экране либо ничего не изменяется, либо старый цвет заменяется на цвет рисования, как при прямой записи). Однако в многоцветных режимах эти операции могут оказаться полезны. Дело в том, что цвета в компьютере кодируются двоичными числами и логические операции выполняются над этими числами *поразрядно*. Например, в 16-цветном режиме с кодировкой цветов

0000 — черный	0101 — сиреневый
0001 — синий	0110 — желтый
0100 — красный	0111 — белый

при рисовании с различными логическими операциями

получаются следующие результаты:

Старый цвет	Цвет рисования	Цвет после рисования с операцией & (логическ. И)	(логическ. ИЛИ)
черный	синий	черный	синий
синий	синий	синий	синий
красный	синий	черный	сиреневый
черный	желтый	черный	желтый
синий	желтый	черный	белый
красный	желтый	красный	желтый

4. Исполнитель Курсор.

Хотя реализация этого исполнителя подробно описана в учебнике, разбираться в ее деталях не обязательно. Вполне достаточно разобраться в командах исполнителя и научиться его использовать. Важно при обсуждении отметить, что, хотя по своей природе исполнители ГраТекс и Курсор различны (первый дан заранее, и его тексты недоступны, а второй написан на школьном алгоритмическом языке специально для этого проекта), в их использовании нет никаких различий.

Домашнее задание.

Прочитать пункты 11.5—11.8.

Урок 20

Тема урока: Графический редактор (продолжение).

Программное обеспечение: Страница *Графический редактор* гипертекста НХЗ.

Цели урока: Ознакомление с общей структурой графического редактора, разбор алгоритма «резинной нити».

План урока:

1. Обсуждение структуры программы.

2. Изменение текущего положения.
 3. Научим компьютер рисовать.
 4. Алгоритм «резиновой нити».
- Домашнее задание.

1. Обсуждение структуры программы.

На этом уроке строится некий «каркас» программы, к которому потом постепенно будут добавляться разные блоки (алгоритмы). Это является широко распространенным способом написания программ (он называется «сверху вниз»): сначала пишется головная часть и отлаживается основная логика программы, а затем дописываются разнообразные функции.

Добавление новой функции к редактору очень просто: пишется соответствующий алгоритм. Кроме этого модифицируется обработка введенного символа: вставляется один или несколько блоков вида

```

|  выбор
|  | при a="...":
|  | ... | вызов нового алгоритма

```

2. Изменение текущего положения.

Первый работающий (но еще не рисующий) вариант программы может быть рассмотрен очень кратко.

Решение упражнения 38. Курсор будет передвигаться в пять раз быстрее, чем при нажатии просто на стрелки.

3. Научим компьютер рисовать.

При рассмотрении следующего варианта программы обратите внимание на конструкцию

лин:=1-лин

Она является часто используемым видом триггера,

в котором при каждом исполнении этой строки значение лин циклически изменяется (0,1,0,1,0,...).

После очень краткого обсуждения текста программы надо несколько минут урока уделить практической работе с этим вариантом редактора. Ученики должны убедиться, что программа уже работает, но пользоваться ей еще очень неудобно.

4. Алгоритм «резиновой нити».

Завершается урок разбором принципа интерфейса «резиновой нити». Алгоритм, реализующий «резиновую нить», имеет внутреннюю структуру, которая будет использована в следующем уроке при реализации аналогичных алгоритмов рисования прямоугольника, эллипса, окружности.

В зависимости от скорости усвоения материала возможно и перераспределение материала по урокам. Возможен перенос части материала из следующего урока в этот с целью дать учащимся больше времени для самостоятельной работы с машиной. Обратный перенос хотя и возможен, но не желателен, так как замечено, что чисто теоретические познания в программировании без соответствующей практики бесполезны.

Домашнее задание.

Прочитать пункты 11.9 и 11.10.

Урок 21

Тема урока: Графический редактор (окончание).

Программное обеспечение: Страница *Графический редактор* гипертекста НХЗ.

Цель урока: Закрепление пройденного материала.

План урока:

1. Разбор алгоритма рисования прямоугольника.

2. Обсуждение алгоритмов рисования эллипса и окружности.

3. Практическая работа по реализации алгоритмов.

Домашнее задание.

Этот урок — в основном самостоятельная работа учащихся на закрепление пройденного материала.

1. Разбор алгоритма рисования прямоугольника.

В тексте учебника описывается реализация алгоритма рисования прямоугольника. Этот алгоритм должен быть разобран и опробован в начале урока.

2. Обсуждение алгоритмов рисования эллипса и окружности.

Далее в учебнике рассказываются основные идеи алгоритмов рисования эллипса и окружности. Реализовать их предоставляется самим учащимся, однако перед началом самостоятельной работы эти идеи следует обсудить.

Хотя в этом уроке реализуется много новых алгоритмов, это не должно вызывать затруднений, поскольку они все аналогичны алгоритму рисования «резиновой нити», разобранному на прошлом уроке.

3. Практическая работа по реализации алгоритмов.

Тексты алгоритмов рез_овал и рез_круг приведены ниже. Кроме того, в гипертексте учащимся предоставляется возможность посмотреть текст графического редактора, попробовать его изменить и увидеть, что из этого получится.

алг рез_овал (арг лог закрашивать)

нач лит а, сим s

цел x0,y0 | позиция «закрепленного» конца

цел dx,dy | смещение

x0:=x; y0:=y

s:=лог_оп; лог_оп:=''

курсор_спрятать

поз(x0,y0) | нарисуем линию из начала

контур(0,0) | нулевого размера

курсор_показать

нц

a:=getkey; выч_смещ(a,dx,dy)

поз(цел((x+x0)/2),цел((y+y0)/2))

эллипс(цел(abs((x-x0)/2)),цел(abs((y-y0)/2)))

x:=x+dx;y:=y+dy

курсор_поз(x,y)

поз(цел((x+x0)/2),цел((y+y0)/2))

эллипс(цел(abs((x-x0)/2)),цел(abs((y-y0)/2)))

кц при a="Enter" или a="Esc Esc"

курсор_спрятать

поз(цел((x+x0)/2),цел((y+y0)/2))

эллипс(цел(abs((x-x0)/2)),цел(abs((y-y0)/2)))

если a="Enter"

то лог_оп:='i'

поз(цел((x+x0)/2),цел((y+y0)/2))

если закрашивать

то

овал(цел(abs((x-x0)/2)),цел(abs((y-y0)/2)))

иначе

эллипс(цел(abs((x-x0)/2)),цел(abs((y-y0)/2)))

все

все

лог_оп:=s; курсор_показать

кон

алг рез_круг (арг лог закрашивать)

нач лит а, сим s

цел x0,y0 | позиция «закрепленного» конца

цел dx,dy | смещение

x0:=x; y0:=y

s:=лог_оп; лог_оп:=''

курсор_спрятать

поз(x0,y0) | нарисуем окружность

окружность(0) | нулевого размера

курсор_показать

Урок 22

Тема урока: Информационная модель в игре «Мудрый Крот».

Программное обеспечение: Страница *Игра «Мудрый Крот»* гипертекста НХЗ.

Цель урока: Ознакомление с информационной моделью в игре «Мудрый Крот».

План урока:

1. Знакомство с игрой «Мудрый Крот».
2. Обсуждение модели игрового поля.
3. Можно ли сделать шаг вправо?
- 4*. Упакованное хранение лабиринтов.

Домашнее задание.

1. Знакомство с игрой «Мудрый Крот».

Игра «Мудрый Крот» — это компьютерная головоломка, заключающаяся в следующем. Играющий, управляя Кротом в одном из множества подземных двухмерных лабиринтов, должен собрать ящики, разбросанные по лабиринту, переместив их на склад. Можно лишь толкать перед собой один ящик. Игра «Мудрый Крот» реализована в системе КуМир и насчитывает 60 лабиринтов. У этой головоломки есть зарубежный аналог под названием Soukovan, реализованный в свое время на ПЭВМ Yamaha корпорацией ASCII (которая, кстати, не имеет никакого отношения к коду ASCII).

2. Обсуждение модели игрового поля.

На самом деле в компьютерной реализации поле кодируется таблицей `цел таб поле [0:30,0:20]`. Числа взяты с запасом, чтобы в эту таблицу помещался самый большой лабиринт.

Решение упражнений:

40. Разумеется, в описанной информационной модели не хватает самого главного: в ней нет самого Кро-

нц

```
a:=getkey; выч_смещ(a,dx,dy)
поз(x0,y0)
окружность(цел(sqrt((x-x0)**2+1.88*(y-y0)**2)))
x:=x+dx;y:=y+dy
курсор_поз(x,y)
поз(x0,y0)
окружность(цел(sqrt((x-x0)**2+1.88*(y-y0)**2)))
кц_при a="Enter" или a="Esc Esc"
курсор_спрятать
поз(x0,y0)
окружность(цел(sqrt((x-x0)**2+1.88*(y-y0)**2)))
```

```
если a="Enter"
то лог_оп:='i'
поз(x0,y0)
если закрашивать
то круг(цел(sqrt((x-x0)**2+1.88*(y-y0)**2)))
иначе
окружность(цел(sqrt((x-x0)**2+1.88*(y-y0)**2)))
все
все
лог_оп:=s; курсор_показать
```

кон

При выполнении практической работы ученикам предоставляются тексты алгоритмов `рез_овал` и `рез_круг`, которые являются просто копией текста `рез_прял`. Эти тексты требуется исправить так, чтобы они соответствовали своему названию.

Полный текст графического редактора состоит из нескольких фрагментов. В каждый момент времени доступен только один фрагмент, для перехода к следующему (или предыдущему) надо использовать команду `[Esc] →` (или `[Esc] ←` соответственно).

Домашнее задание.

Прочитать § 12.

та. Нетрудно исправить этот недостаток. Например, добавим к модели два целых числа, задающих положение Крота на поле:

цел x, y

41. Лабиринт считается пройденным, когда на поле не осталось ни одного ящика, стоящего не на складе. Заметим, что для проверки этого условия требуется просмотреть все поле, что займет довольно много времени. Однако в построенной информационной модели другого способа нет! Для того чтобы ускорить проверку, надо дополнить модель, например, переменной

цел остаток

в которой будет храниться число ящиков, которые еще осталось передвинуть.

3. Можно ли сделать шаг вправо?

Разобранный в пособии алгоритм используется для выполнения четырех команд перемещения Крота. При этом он вызывается со следующими аргументами:

пойти(-1, 0) = влево
пойти(1, 0) = вправо
пойти(0, -1) = вверх
пойти(0, 1) = вниз

4*. Упакованное хранение лабиринтов.

В хорошо подготовленных классах в конце урока можно дать задачу по тексту программы восстановить способ упаковки алгоритмов. Он заключается в следующем.

Лабиринт кодируется последовательностью битов (алгоритм-функция следующий бит при каждом очередном обращении выбирает из последовательности очередной бит и возвращает его в качестве свое-

го значения). Упакованное представление лабиринта состоит из пар

повторитель значение

где *повторитель* указывает, сколько раз надо повторить *значение* при построении лабиринта. Оба числа содержат переменное число битов. *Повторитель* может состоять из 0 (это означает, что *значение* надо повторить 1 раз), либо из 1, за которой следует три двоичные цифры (при этом $1x_1x_2x_3$ означает, что *значение* надо повторить $2^2x_1 + 2x_2 + x_3 + 2$ раз). *Значение* кодирует элементы лабиринта следующим образом:

00 пустая клетка (проход)
01 клетка, занятая стеной
10 клетка с ящиком
110 пустая позиция на складе
111 ящик, размещенный на складе

После заполнения одного ряда лабиринта переход на следующий выполняется автоматически (это реализовано в алгоритме вывести, в котором заодно считается число позиций на складе, которые еще не заполнены).

Домашнее задание.

Прочитать введение к § 13 и пункт 13.1.

Глава 4. КОМПЬЮТЕРЫ В ОБУЧЕНИИ

Система ПланиМир

В качестве примера использования компьютеров в обучении взята система геометрических построений ПланиМир. Система ПланиМир позволяет при помощи мышки ставить точки, проводить отрезки, прямые и окружности, измерять расстояния и углы и т.д. Таким образом, ученику предоставляется среда, в которой можно делать любые аналоги построений на бумаге циркулем и линейкой. Рассмотрим кратко, чем использование компьютера удобнее традиционных построений на бумаге.

Построение чертежей

Первая особенность ПланиМира — возможность строить аккуратные чертежи, даже не имея никаких способностей к черчению. Достаточно отметить на экране две точки и дать команду «провести отрезок», и компьютер сам нарисует нужный отрезок и даже даст ему имя (которое, впрочем, можно впоследствии легко изменить). Построение прямых, лучей и окружностей не сложнее (все построения достаточно подробно описаны в книге для ученика). Таким образом, ПланиМир дает возможность заниматься геометрией, не отвлекаясь на разные второстепенные проблемы.

Базовым объектом системы является точка. Для «построения» точки достаточно подвести курсор в требуемое место экрана и нажать левую кнопку мышки. Все остальные фигуры являются производными от точек. Так, например, прямая проводится

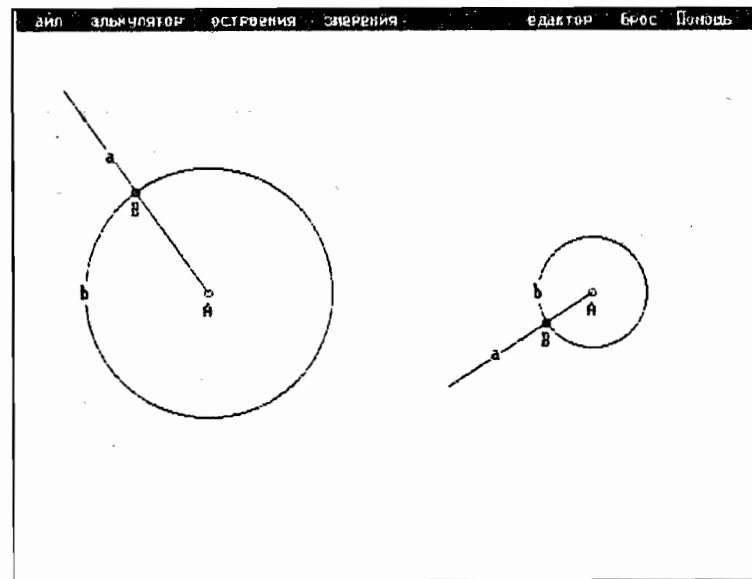


Рис. 1. Трансформация чертежа в ПланиМире

через две точки, а отрезок соединяет их. Способ выполнения построений совпадает с принятым практически во всех современных системах порядком действий: сначала отмечаются объекты, над которыми производится действие (в ПланиМире объекты отмечаются правой кнопкой мыши), а потом дается команда на выполнение действия (путем выбора из меню или нажатием ее клавиатурного эквивалента).

Трансформация чертежа в ПланиМире

Вторая, более важная и более интересная особенность ПланиМира — это возможность трансформировать уже готовый чертеж, двигая одну из исходных точек или прямых. Важно отметить, что все построения при этом сохраняются. Например, поставим две точки A и B , проведем через них прямую и построим окружность с центром в точке A и ра-

диусом AB (рис. 1, слева). Теперь, если мы будем перемещать точку B , то чертеж изменится (рис. 1, справа), но луч по-прежнему будет проходить через точки A и B , а радиус окружности по-прежнему будет равен AB (хотя это будет уже другое число).

Каковы же применения возможности трансформации чертежа? Во-первых, не надо задумываться о положении базовых точек. При построениях на бумаге часто оказывается, что в одном месте чертежа точки стоят слишком плотно и ничего нельзя разобрать, хотя в другом месте еще полно свободного места. В результате приходится все перерисовывать заново. В ПланиМире это не так — всегда можно подвинуть исходные точки так, чтобы чертеж был наглядным.

Во-вторых, появляется возможность проверить построение, не вдаваясь в его детали. В самом деле, пусть ученик построил биссектрису угла, но использовал при этом какой-то непонятный вам метод. Построенный им луч на чертеже действительно похож на биссектрису, но, может быть, это получилось случайно и этот метод годится только для угла в 60 градусов? Можно, конечно, объяснить ученику, что биссектрису строят не так, что есть другой, «правильный» метод построения, но это непедагогично и отучает детей от самостоятельности.

При использовании ПланиМира такой проблемы нет. Достаточно трансформировать чертеж, изменяя величину угла и положение его вершины. Если построенный учеником луч при всех модификациях чертежа по-прежнему делит угол пополам, значит, построение правильное, каким бы странным оно ни казалось. Разумеется, ученик может (и должен) сам попробовать трансформировать чертеж, прежде чем показывать его учителю.

И наконец, возможность трансформации чертежа позволяет организовать самостоятельную поисковую работу учеников, что вообще невозможно при использовании традиционных средств. Известно, что факт, «открытый» самостоятельно, гораздо лучше запоминается, чем услышанный или прочитанный в книжке. Но попробуйте заставить учеников построить на уроке десяток аккуратных чертежей! В системе ПланиМир учитель вполне может дать следующее задание: «Посмотрите, в каком отношении медианы делятся их точкой пересечения». Выполнив необходимые построения и проведя измерения, ученики видят, что это отношение равно 2:1. Может быть, это случайность? Нет, при всех трансформациях чертежа отношение сохраняется.

Важно отметить, что трансформация чертежа не является строгим математическим доказательством (хотя она может являться доказательством при некоторых условиях, но обоснование этого выходит далеко за рамки школьной программы). Таким образом, полученный результат все равно надо доказывать. Но для учеников это уже не будет просто сухим утверждением из книжки.

Урок 23

Тема урока: Возможности и общее назначение системы ПланиМир.

Программное обеспечение: Страницы *Демонстрация возможностей системы* и *Свободная работа в ПланиМире* из гипертекста по геометрии (PLANIMIR).

Цели урока: Первоначальное знакомство с системой ПланиМир, освоение команд запуска системы и выхода из нее.

План урока:

1. Обсуждение домашнего задания.
2. Практическая работа в системе ПланиМир.
Домашнее задание.

1. Обсуждение домашнего задания.

Начало урока следует посвятить краткому обсуждению начала § 13.

2. Практическая работа в системе ПланиМир.

После этого можно сразу переходить к практическому знакомству с системой ПланиМир (пункт 13.1). Материал в книге для ученика достаточно подробно описывает все действия, выполнение которых должны освоить ученики.

Домашнее задание.

Прочитать пункты 13.2—13.5.

Урок 24

Тема урока: Построения в ПланиМире.

Программное обеспечение: Страницы *Свободная работа в ПланиМире* и *Живые чертежи* гипертекста PLANIMIR.

Цель урока: Освоение команд построения точек и прямых, их пометки и перемещения в системе ПланиМир.

План урока:

1. Обсуждение домашнего задания.
2. Практическая работа в системе ПланиМир.
- 3*. Рассмотрение реальной модели работы с помеченными объектами.
Домашнее задание.

1. Обсуждение домашнего задания.

Урок рекомендуется построить на базе материала пунктов 13.2—13.5. Поскольку этот материал

не содержит никаких сложных теоретических понятий, на его обсуждение достаточно отвести буквально несколько минут.

2. Практическая работа в системе ПланиМир.

Основную часть урока должна занять практическая работа на компьютере. Ученики должны освоить «построение» точек, их пометку и перемещение, выполняя задание из пункта 13.3. Аналогичные задания учитель может дать при изучении построения прямой.

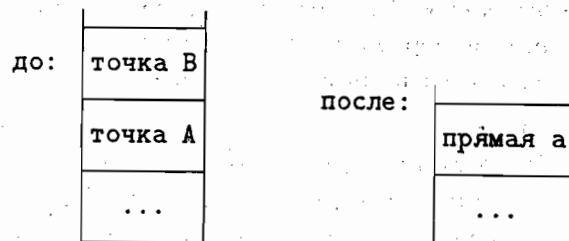
3*. Рассмотрение реальной модели работы с помеченными объектами.

В книге для ученика описана наиболее примитивная модель работы с ПланиМиром. Ее вполне достаточно для работы с системой, однако в хорошо подготовленных классах можно рассмотреть реальную модель работы с помеченными объектами.

В основу этой модели положено понятие *стека*. Стек работает по принципу «пришел последним — ушел первым» (last in — first out), т.е. объект, помещенный в стек последним, достается первым. Хорошим примером стека служит стопка книг на столе — на вершине стопки окажется книга, положенная последней, а для того, чтобы достать книгу из середины стопки, надо сначала снять все верхние.

Объекты в ПланиМире помещаются в стек в порядке их пометки, и в стеке с ними могут выполняться операции. Аргументами операции будут несколько верхних (т.е. помещенных в стек последними) объектов. После выполнения операции они удаляются из стека, а результат выполнения операции заносится в стек. Например, пусть последними были помечены (и значит, помещены в стек) две точки — сначала А, потом В. Тогда при выполнении команды Прямая состояние стека изменится следую-

щим образом:



Из стека были извлечены аргументы (точка А и точка В) и помещен результат (прямая а). Точно так же выполняются операции над числами и другими объектами. Использование стека значительно ускоряет построения.

В качестве упражнения можно разобрать, как будет изменяться состояние стека помеченных объектов при выполнении следующего построения (нахождение точки пересечения двух отрезков, заданных концами):

- отметить точку А
- отметить точку В
- построить отрезок
- отметить точку С
- отметить точку D
- построить отрезок
- пересечь отрезки

Домашнее задание.

Прочитать пункты 13.6—13.8.

Урок 25

Тема урока: Построения в ПланиМире (продолжение).

Программное обеспечение: Страница *Свободная работа в ПланиМире* гипертекста PLANIMIR.

Цели урока: Ознакомление с понятиями объектов-родителей и объектов-детей, освоение команд построения отрезков, лучей, окружностей и точек на объектах в системе ПланиМир.

План урока:

1. Обсуждение домашнего задания.
 2. Практическая работа в системе ПланиМир.
 3. Построение точек, лежащих на объектах.
- Домашнее задание.

1. Обсуждение домашнего задания.

Материал для урока — пункты 13.6—13.9. Важное понятие, которое надо обсудить в начале урока 25, — *родители и дети* (пункт 13.6). Далее надо кратко рассмотреть материал пунктов 13.7 и 13.8, после чего переходить к практической работе.

2. Практическая работа в системе ПланиМир.

Построение отрезка (13.7) практически ничем не отличается от построения прямой и не должно вызывать никаких трудностей. При построении луча и окружности (13.8) важен порядок, в котором помечаются точки. Хотя этот порядок вполне отвечает интуитивным представлениям, возможны ошибки (особенно при выполнении серии построений). В этом случае можно забежать немного вперед и рассмотреть команду Откатить, описанную в пункте 13.11.

3. Построение точек, лежащих на объектах.

Перед обсуждением материала пункта 13.9 рекомендуется подвести учеников к необходимости построения связанных точек. Можно обсудить, хватает ли рассмотренных средств для решения какой-нибудь простой задачи (например, построения прямого угла). После того как в ходе обсуждения будут предложены команды «построить точку на объекте»

и «пересечь», можно рассказать, как эти команды выполнять в ПланиМире. Остаток урока следует посвятить практической работе на компьютере по закреплению этих команд.

Домашнее задание.

Прочитать пункты 13.10—13.11.

Урок 26

Тема урока: Измерения и вычисления.

Программное обеспечение: Страница *Свободная работа в ПланиМире* гипертекста PLANIMIR.

Цель урока: Формирование навыков измерений и вычислений в системе ПланиМир.

План урока:

1. Обсуждение домашнего задания.
2. Практическая работа в системе ПланиМир.

Домашнее задание.

1. Обсуждение домашнего задания.

При измерениях у учеников могут возникнуть трудности, связанные с тем, что результат измерения сам является объектом, который можно помечать, перемещать, удалять и т.д.

2. Практическая работа в системе ПланиМир.

Основная часть урока должна пройти в форме самостоятельного решения простых упражнений по отдельным командам. Например, для освоения команды Переименовать можно предложить построить треугольник XYZ . ПланиМир при построении точек сам дает им имена, начиная по порядку с начала алфавита. У этой задачи есть «нечестное» решение: поставить сначала 23 точки (они получают имена от A до W), а потом уже строить треугольник. Это решение можно также принять, но потребовать по край-

ней мере спрятать все лишние точки (и тогда это получится задача на команду Спрятать).

Точки A , B и C лежат на одной прямой, если угол $\angle ABC$ равен 0° или 180° . Не нужно требовать от учеников получения на экране абсолютно точного результата (из-за погрешностей компьютерной арифметики это может оказаться очень сложным), однако важно, чтобы они могли сформулировать, какого результата хотят добиться.

Домашнее задание.

Прочитать пункт 13.12.

Урок 27

Тема урока: Решение задач на построение.

Программное обеспечение: Страница *Геометрический практикум* гипертекста PLANIMIR.

Цель урока: Закрепление навыков работы в системе ПланиМир.

План урока:

1. Разбор готовых решений.
2. Самостоятельное решение задач.

Домашнее задание.

1. Разбор готовых решений.

В начале урока следует разобрать и обсудить задачи, решение которых приведено в пункте 13.12. Рекомендуется использовать при этом приведенные в гипертексте демонстрации. Однако если окажется, что ученикам трудно переключить внимание от компьютера к общему обсуждению, можно заменить работу в гипертексте разбором задач у доски (с последующим самостоятельным просмотром демонстраций).

2. Самостоятельное решение задач.

В пункте 13.12 приведено 8 задач для самостоятельного решения. Учитель может выбрать из них необходимое для заполнения урока число задач либо предложить свои.

Решение упражнений:

42. Проведем окружность с центром в точке A и радиусом AC . Проведем окружность с центром в точке B и радиусом BC . Найдем их точку пересечения D , отличную от C . Проведем отрезки AD и BD . Треугольник ABD — искомый.

Доказательство. По построению $AC = AD$, $BC = BD$, AB — общая сторона. Значит, треугольники равны по трем сторонам.

43. Проведем окружность с центром в точке C и радиусом AB . Проведем окружность с центром в точке D и радиусом AB . Найдем их точку пересечения O . Проведем окружность s с центром в точке O и радиусом AB . Окружность s — искомая.

Доказательство. Точки B и C удалены от точки O на расстояние AB , поэтому окружность s проходит через них.

44 и 45. Для решения этих задач нужно вначале построить угол, равный данному. Дальнейшие построения вполне очевидны.

46. Проведем биссектрису угла BAC . Проведем биссектрису угла ABC . Найдем их точку пересечения O . Опустим перпендикуляр OX на сторону AB . Проведем окружность s с центром в точке O и радиусом OX . Окружность s — искомая.

47. Отметим точки B и C на различных сторонах угла. Проведем биссектрису AD угла A . Проведем биссектрису AE угла BAD . Проведем биссектрису AF угла DAC .

48. Найдем середину D отрезка AB . Соединим ее с вершиной C . Найдем середину E отрезка BC . Соединим ее с вершиной A . Найдем середину F отрезка AC . Соединим ее с вершиной B . Отрезки AE , BF , CD — искомые.

49. Проведем произвольную прямую. Отметим на ней точку X и отложим от нее отрезок XU , равный AB . Найдем середину U отрезка XU . Проведем окружность с центром Y и радиусом CD . Проведем окружность с центром U и радиусом EF . Найдем их точку пересечения Z . Соединим точки Z и X , Z и Y . Треугольник XUZ — искомый.

Домашнее задание.

Прочитать пункт 13.13.

Урок 28

Тема урока: Решение задач на построение (продолжение).

Программное обеспечение: Страница *Геометрический практикум* гипертекста PLANIMIR.

Цель урока: Закрепление навыков работы в системе ПланиМир.

План урока:

1. Разбор готовых решений.
2. Самостоятельное решение задач.

Домашнее задание.

Форма проведения этого урока ничем не отличается от описанной выше для урока 27.

Решение упражнений:

50. Опустим перпендикуляр AF из точки A на сторону BC . Опустим перпендикуляр BE из точки B на сторону AC . Опустим перпендикуляр CH из точки C на сторону AB . Отрезки AF , BE и CH — искомые.

51. Найдем середину E отрезка AB . Найдем середину F отрезка BC . Проведем через точку E прямую a , перпендикулярную AB . Проведем через точку F прямую b , перпендикулярную BC . Найдем точку O пересечения прямых a и b . Проведем окружность c с центром O и радиусом OA . Окружность c — искомая.

52. Отметим произвольную точку C на прямой b . Восстановим перпендикуляр c из C к b . Отложим от C на c отрезки CD и CE , равные AB . Проведем через точку D прямую d , параллельную b . Найдем ее точку пересечения F с прямой a . Проведем через точку E прямую e , параллельную b . Найдем ее точку пересечения G с прямой a . Точки F и G — искомые.

53. Найдем середину C отрезка AB . Восстановим из точки C перпендикуляр h к AB . Найдем точку O пересечения h и a . Точка O — искомая.

54а. Проведем произвольную прямую. Отметим на ней точку X и отложим от нее отрезок XW , равный AB . Отложим от точки W отрезок WP , равный AB . Проведем окружность с центром в точке X и радиусом EF . Проведем окружность с центром в точке P и радиусом CD . Найдем их точку пересечения Z . Проведем отрезок WZ . Проведем через Z прямую a , параллельную XW . Проведем через X прямую b , параллельную ZW . Найдем их точку пересечения Y . Проведем отрезки YZ и YX . Четырехугольник $WXYZ$ — искомый.

54б. Построим угол X , равный данному. Отложим на одной стороне угла отрезок WX , равный AB , а на другой стороне угла — отрезок XU , равный CD . Проведем через точку W прямую, параллельную XU . Проведем через точку U прямую, параллельную XW . Найдем их точку пересечения Z . Че-

тырехугольник $WXYZ$ — искомый.

54в. Отложим отрезок EL , равный AB , на одной из сторон угла. Найдем середину W отрезка EL . Отложим отрезок EM , равный CD , на другой стороне угла. Найдем середину X отрезка EM . Проведем прямую a через точки W и E . Отложим на ней от точки E отрезок EY , равный EW . Проведем отрезок XU . Проведем прямую b через точки X и E . Отложим на ней от точки E отрезок EZ , равный EX . Проведем отрезок ZW . Проведем отрезки WX и YZ . Четырехугольник $WXYZ$ — искомый.

55а. Проведем биссектрису a угла A . Отложим на ней отрезок WU , равный BC . Проведем через точку U прямую l параллельно одной из сторон угла и найдем точку X , в которой она пересекает другую сторону угла. Проведем через точку U прямую m параллельно WX и найдем точку Z , в которой она пересекает сторону угла. Четырехугольник $WXYZ$ — искомый.

55б. Построим угол W , равный данному. Построим произвольную окружность с центром в точке W . Пусть M и N — точки ее пересечения со сторонами угла. Проведем прямую через точки M и N и отложим на ней отрезок $MP = BC$. Проведем через точку P прямую a параллельно стороне угла и найдем точку X , в которой она пересекает другую сторону угла. Нахождение двух остальных вершин ромба уже не представляет никаких сложностей.

55в. Проведем произвольную прямую. Отметим на ней точку W и отложим от нее отрезок WX , равный AB . Построим окружность a с центром в точке W и радиусом CD . Построим окружность b с центром в точке X и радиусом AB . Найдем их точку пересечения Y . Построим окружность c с центром в точке W

и радиусом AB . Построим окружность d с центром в точке Y и радиусом AB . Найдем их точку пересечения Z . Проведем отрезки WZ и WY . Четырехугольник $WXYZ$ — искомый.

55г. Проведем произвольную прямую. Отметим на ней точку W . Отложим отрезок WY , равный AB . Найдем середину K отрезка WY . Восстановим перпендикуляр c из K к WY , на нем отложим отрезок KL , равный CD . Найдем середину X отрезка KL . От K на c отложим отрезок KZ , равный KX . Четырехугольник $WXYZ$ — искомый.

Домашнее задание.

Повторить § 8 (глава 2). Прочитать пункты 14.1 и 14.2 из § 14.

Урок 29

Тема урока: Модель чертежа в ПланиМире.

Программное обеспечение: Страницы *Демонстрация возможностей системы* и *Свободная работа в ПланиМире* гипертекста PLANIMIR.

Цель урока: Ознакомление с моделью чертежа в ПланиМире.

План урока:

1. Информационная модель рисунка на поле Чертежника (повторение).
 2. Обсуждение модели чертежа в ПланиМире.
 3. Практическая работа в системе ПланиМир.
- Домашнее задание.

1. Информационная модель рисунка на поле Чертежника.

Идея хранения чертежа в виде линейной программы на школьном алгоритмическом языке (т.е.

в виде протокола его построения) сама по себе довольно сложна. Начать урок лучше с повторения материала параграфа 8 («Информационная модель рисунка на поле Чертежника»). Приведенную там модель можно рассматривать как описание рисунка, а можно — как сокращенную запись алгоритма рисования. В ПланиМире принята полная форма записи алгоритма построения чертежа (просто для того, чтобы не придумывать новые обозначения для записи вспомогательных алгоритмов и управляющих конструкций).

2. Обсуждение модели чертежа в ПланиМире.

Важной особенностью команд исполнителя PLANE является то, что в качестве их аргументов могут быть использованы только константы (вещественные, логические и литерные). Использование имен переменных недопустимо, кроме специально оговоренного случая со вспомогательными алгоритмами. Однако не имеет смысла специально акцентировать внимание учеников на этом — в рамках данного курса потребность в использовании переменных вряд ли возникнет.

3. Практическая работа в системе ПланиМир.

Как обычно, результаты обсуждения должны быть закреплены практической работой в системе ПланиМир.

Домашнее задание.

Прочитать пункты 14.3 и 14.4.

Урок 30

Тема урока: Команды исполнителя PLANE.

Программное обеспечение: Страницы *Свободная работа в ПланиМире* и *Программа, составленная вручную* гипертекста PLANIMIR.

Цели урока: Ознакомление с командами исполнителя PLANE, формирование навыков чтения и модификации протоколов построения.

План урока:

1. Обсуждение списка команд исполнителя PLANE.

2. Практическая работа в системе ПланиМир.

3. Использование составных команд в протоколе.
Домашнее задание.

1. Обсуждение команд исполнителя PLANE.

Урок следует начать с разбора пункта 14.3 («Список команд исполнителя PLANE»). Основная задача — ознакомиться с командами исполнителя PLANE и научиться легко разбираться в протоколах и их модифицировать. Это необходимо для освоения работы со вспомогательными алгоритмами.

2. Практическая работа в системе ПланиМир.

Одна из задач, которые можно поставить перед учениками, — выяснить смысл параметра t в командах «точка на объекте». После решения этой задачи ученики должны сами обнаружить «нечестный» способ деления отрезка пополам при помощи команды

точка на отрезке("С", да, да, "а", 0.5)

3. Составные команды в протоколе.

К сожалению, упомянутое выше ограничение на использование имен переменных не позволяет написать алгоритм, ставящий N точек в цикле. Приведенный в учебнике пример алгоритма слишком прост и не очень интересен, и на его рассмотрение не стоит тратить много времени.

Домашнее задание.

Прочитать пункт 14.5.

Урок 31

Тема урока: Вспомогательные алгоритмы.

Программное обеспечение: Страницы *Свободная работа в ПланиМире* и *Подпрограммы в задачах на построение* гипертекста PLANIMIR.

Цель урока: Ознакомление с процессом создания вспомогательных алгоритмов в ПланиМире и с их использованием.

План урока:

1. Обсуждение домашнего задания.

2. Самостоятельное создание вспомогательных алгоритмов.

Домашнее задание.

1. Обсуждение домашнего задания.

Урок строится по материалам пункта «Вспомогательные алгоритмы» (14.5). В начале урока следует разобрать приведенный в книге для ученика пример, после чего можно переходить к практической работе в ПланиМире.

2. Самостоятельное создание вспомогательных алгоритмов.

Создание вспомогательных алгоритмов — занятие достаточно сложное и может быть полностью освоено только наиболее подготовленными учащимися. Однако овладеть использованием уже созданных алгоритмов необходимо всем (это отражает общую ситуацию: программы пишутся профессионалами, а используются всеми).

Первая задача, которую можно поставить перед учениками, — создание алгоритма построения медиан треугольника (треугольник задается тремя своими вершинами). Для решения этой задачи потребуется использовать готовый алгоритм середина.

Домашнее задание.

Прочитать пункт 14.6.

Урок 32

Тема урока: Конференция.

Программное обеспечение: Страницы Подпрограммы в задачах на построение и Конференция гипертекста PLANIMIR.

Цели урока: Закрепление навыков работы с системой ПланиМир, повторение пройденного материала.

План урока:

1. Обсуждение домашнего задания.
2. Решение задач в ПланиМире.
3. Обзор модуля «Информационные модели».

1. Обсуждение домашнего задания.

Урок следует начать с разбора задачи «Теорема о пропорциональных отрезках» (14.6). Решение упражнения 56 может быть легко получено из того факта, что $(abc)/(de) = ((ab)/d)c/e = (yc)/e$, где $y = (ab)/d$.

2. Решение задач в ПланиМире.

Далее следует переходить к самостоятельному решению задач в системе ПланиМир. Возможные задачи:

Задача. Постройте трапецию $WXYZ$ по основаниям $WZ = AB$, $XY = CD$ и боковым сторонам $WX = EF$, $YZ = GH$.

Решение. Проведем произвольную прямую. Отметим на ней точку W . Отложим отрезок WZ , равный AB . Отложим отрезок TZ , равный CD , в сторону A (так, чтобы точка T лежала между W и Z). Проведем окружность с центром в точке W и радиусом EF . Проведем окружность с центром в точке T и

радиусом GH . Найдем их точку пересечения X . Проведем через точку X прямую a , параллельную WZ . Отложим на ней отрезок XY , равный CD . Проведем отрезок YZ . Четырехугольник $WXYZ$ — искомый.

Задача. Постройте трапецию $WXYZ$ по основаниям $WZ = AB$, $XY = CD$ и диагоналям $WY = EF$, $XZ = GH$.

Решение. Проведем произвольную прямую. Отметим на ней точку Y . Отложим отрезок XY , равный CD . Отложим отрезок YT , равный AB , в сторону, противоположную X (так, чтобы точка Y была между X и T). Проведем окружность с центром в точке X и радиусом GH . Проведем окружность с центром в точке T и радиусом EF . Найдем их точку пересечения Z . Проведем через точку Z прямую a , параллельную XY . Отложим на ней отрезок WZ , равный AB . Проведем отрезки WX и YZ . Четырехугольник $WXYZ$ — искомый.

Задача. Постройте треугольник XYZ по данным серединам его сторон A , B , C .

Решение. Проведем через A прямую a , параллельную BC . Проведем через B прямую b , параллельную AC . Проведем через C прямую c , параллельную AB . Найдем точку пересечения X прямых a и b . Найдем точку пересечения Y прямых b и c . Найдем точку пересечения Z прямых a и c . Треугольник XYZ — искомый.

Подготовленную среду для этих задач можно найти, запустив гипертекст и войдя в пункт Конференция на головной странице.

3. Обзор модуля «Информационные модели».

Завершить урок следует обзором содержания модуля 10 класса, построенным в виде кратких докладов учеников по заранее распределенным темам.

Оглавление

Глава 1. ПЕРЕДАЧА ИНФОРМАЦИИ.....	3
Урок 1. Передача информации между людьми..	—
Дополнительный материал	4
Урок 2. Двоичное кодирование (повторение).....	6
Урок 3. Протоколы передачи.....	7
Урок 4. Компьютерные сети.....	10
Дополнительный материал	15
Урок 5. Практическая работа с сетью	17
Урок 6. Что такое модем	20
Глава 2. ИНФОРМАЦИОННЫЕ МОДЕЛИ.....	23
Урок 7. Понятие информационной модели	25
Урок 8. Простейшая модель кинозала	27
Урок 9. Конкурирующие расширения модели ..	30
Урок 10. Окончательная модель кинозала	31
Урок 11. Информационная модель транспортной сети	32
Урок 12. Модели геометрической информации ..	36
Урок 13. Модель обстановки на поле Робота ...	38
Урок 14. Модель рисунка на поле Чертежника ..	39
Уроки 15—16. Решение задач на построение информационных моделей	40
Глава 3. КАК УСТРОЕНЫ КОМПЬЮТЕРНЫЕ ПРОГРАММЫ	41
Урок 17. «Записная книжка». Пример базы данных	—
Урок 18. «Записная книжка» (продолжение) ...	43
Урок 19. Графический редактор (начало).....	45
Урок 20. Графический редактор (продолжение) ..	47
Урок 21. Графический редактор (окончание) ...	49
Урок 22. Информационная модель в игре «Мудрый Крот»	53

Глава 4. КОМПЬЮТЕРЫ В ОБУЧЕНИИ	56
Система ПланиМир	—
Построение чертежей	—
Трансформация чертежа в ПланиМире	57
Урок 23. Возможности и общее назначение системы ПланиМир	59
Урок 24. Построения в ПланиМире	60
Урок 25. Построения в ПланиМире (продолжение)	62
Урок 26. Измерения и вычисления	64
Урок 27. Решение задач на построение	65
Урок 28. Решение задач на построение (продолжение)	67
Урок 29. Модель чертежа в ПланиМире	70
Урок 30. Команды исполнителя PLANE	71
Урок 31. Вспомогательные алгоритмы.....	73
Урок 32. Конференция	74

Учебное издание

Кушниренко Анатолий Георгиевич
Леонов Александр Георгиевич
Эпиктетов Михаил Геннадьевич
и другие

ИНФОРМАЦИОННАЯ КУЛЬТУРА
Модуль
Класс 10

Методическое пособие

Ответственный редактор *М. Г. Циновская*
Художник обложки *А. В. Кузнецов*
Оригинал-макет подготовил *М. Г. Эпиктетов*
Корректор *Л. А. Александрова*

ЛР № 061622 от 23 сентября 1992 г.
Подписано к печати 15.08.95 г. Формат 84×108 1/32.
Бумага тип. № 2. Гарнитура «Школьная». Печать высокая.
Усл. печ. л. 4,2. Уч.-изд. л. 4,6. Тираж 10 000 экз.
Заказ № 1171.

Издательский дом «Дрофа».
105318, Москва, ул. Щербаковская, 3

Смоленская областная ордена «Знак Почета» типография
им. Смирнова. 214000, г. Смоленск, пр. им. Ю. Гагарина, 2

**Издательский дом «Дрофа»
выпускает учебники и учебные
пособия известных авторов по всем
курсам школьной программы.
Оптовые поставки во все
регионы России.**

Телефоны: (095) 369-99-19, 369-06-53

Факс: (095) 369-06-53