

Театр начинается с вешалки...

...а книга — с предисловия.

Вы любите читать предисловия к научным книжкам? Я — очень. Случается, конечно, что предисловие выполняет в книге исключительно техническую функцию — в нем сообщается о целях, задачах, принятых соглашениях и обозначениях и т.п. Но нередко (а в книгах по программированию — часто) предисловие является единственным местом, где автор имеет возможность поговорить с читателем, пусть о деле, но “за жизнь”. В этом смысле предисловие становится не столько местом встречи профессионалов, сколько личным письмом автора читателю.

О предисловиях к учебникам — отдельный разговор. Школьная учебная литература имеет не только строгие внешние формальные рамки, но и богатые традиции. Опыт показывает, что привычно структурированные и оформленные учебники принимаются (и понимаются) лучше новомодных изысканных изданий. Тем интереснее бывает встретить учебник, в предисловии которого за традиционным “Здравствуйте, ребята! Мы начинаем изучать...” следует текст интересный, содержательный и самоценный.

В этом году в издательстве “Просвещение” вышел учебник “Информатика-6. Алгоритмика”, написанный А.К. Звонкиным, С.К. Ландо и А.Л. Семеновым. По моим представлениям, само издание (хотя это нигде не написано) является стереотипным учебнику “Алгоритмика 5–7”, изданному “Дрофой” почти десять лет назад. Учебник сам по себе просто замечательный, но я бы, наверное, не стал специально обращать на него внимание читателей, если бы не отличный повод перечитать предисловие (также стереотипное).

Это как раз тот случай, когда предисловие, адресованное ученикам, реально помогает учителю в создании мотивации к изучению предмета. Причем дело не ограничивается данным конкретным учебником — те же слова учитель может вспомнить, работая по любому учебнику в любом классе.

“Главное, что отличает специалиста по программированию, — это умение ясно мыслить. Его указания должны быть настолько ясными, чтобы их мог понять даже компьютер. Вот такой ясности мысли мы с вами и будем учиться”.

<...>

“...одна особенность отличает компьютерное мышление от человеческого. Компьютер абсолютно послушен. Часто бывает так, что вы имели в виду одно, а написали другое. Человек в этой ситуации может сам догадаться, что подразумевалось, и сделать все как надо. Компьютер так никогда не поступит. Он всегда выполнит в точности то, что ему указано, несмотря на все очевидные нелепости. Но и этим он помогает нам, так как учит правильно мыслить и четко излагать свои мысли”.

Правда, хорошо написано? Про “правильно мыслить” мне, честно говоря, не очень нравится, а в остальном — просто здорово!

Я рекомендую всем коллегам, у кого нет ни одного издания “Алгоритмики”, обязательно найти эту книгу для своей библиотеки (помимо предисловия, в ней еще есть много чего интересного).

Жалко только, что художники “Просвещения” умудрились вlepить прямо на обложку нелепый (и, вообще говоря, неверный) элемент блок-схемы. С этим придется смириться ☺.

С.Л. Островский, главный редактор

СОДЕРЖАНИЕ НОМЕРА

ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ

В.Ф. Бурмакина, И.Н. Фалина. Как готовиться к тестированию по проверке ИКТ-компетенции школьников. Лекция 4. Описание познавательных деятельностей, составляющих компетентность “управление”. Примеры заданий на выработку этой компетентности 2–7
Контрольная работа № 1 8

ПРОФИЛЬНАЯ ШКОЛА. ЭКЗАМЕНЫ

Е.А. Еремин, А.П. Шестаков. Примерные ответы на профильные билеты 10–22

ТЕЛЕКОММУНИКАЦИИ

А.А. Дуванов. Азы информатики. Выходим в Интернет 23–33

“В МИР ИНФОРМАТИКИ” № 79

Газета для пытливых учеников и их талантливых учителей

Microsoft Excel углубленно

Рекуррентные соотношения в Microsoft Excel 37–38

Школа программирования

Основы программирования на Visual Basic 39–41

“Ломаем” голову

Минимальная сумма 41
Кто поймал Соловья-разбойника? 42

Это полезно знать

Автоматическая память 43–44

Games.exe

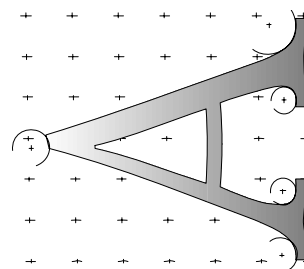
Компьютерные фокусы на отгадывание чисел ... 44–46

КОНКУРС

“Как это делаю я”. Методический конкурс для учителей информатики ... 48

№ 20 (526)

16–31 октября 2006



Методическая газета для учителей информатики

ИНФОРМАТИК

ГАЗЕТА "ИНФОРМАТИКА" И ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ "ПЕРВОЕ СЕНТЯБРЯ"
(лицензия А225682, № 020503 от 19.07.2006)
ПРЕДСТАВЛЯЮТ НОВЫЙ ДИСТАНЦИОННЫЙ КУРС
ПОВЫШЕНИЯ КВАЛИФИКАЦИИ ДЛЯ УЧИТЕЛЕЙ ИНФОРМАТИКИ

В.Ф. Бурмакина, И.Н. Фалина

Как готовиться к тестированию по проверке ИКТ-компетентности школьников

Учебный план

№ газеты "Информатика"	Учебные материалы
17/2006	Лекция 1. Цели и задачи проекта по оценке ИКТ-компетентности девятиклассников. Цели и метод тестирования. Описание когнитивных компетентностей, оцениваемых тестом. Структура теста.
18/2006	Лекция 2. Описание познавательных деятельности, составляющих компетентность "определение информации". Примеры заданий на выработку этого умения.
19/2006	Лекция 3. Описание познавательных деятельности, составляющих компетентность "доступ к информации". Как выбрать информационный ресурс, соответствующий заданным критериям? Какие стратегии можно и нужно использовать для поиска информации в многочисленных, часто противоречивых, источниках?
20/2006	Лекция 4. Описание познавательных деятельности, составляющих компетентность "управление". Могут ли ваши ученики разработать самостоятельно или выбрать из предложенных такое представление исходной информации, которое будет наиболее понятно конкретной аудитории? Понимают ли ваши ученики, что с конфиденциальной информацией надо обращаться в соответствии с определенными нормами? Контрольная работа № 1.
21/2006	Лекция 5. Описание познавательных деятельности, составляющих компетентность "оценка информации". Школьнику предлагается несколько информационных источников (например, статьи, сайты и т.п.), из которых он должен выбрать один, наиболее полно удовлетворяющий заданной потребности.
22/2006	Лекция 6. Описание познавательных деятельности, составляющих компетентность "интеграция информации". Если школьник умеет анализировать различные источники по одной и той же теме и на основе этой деятельности создавать новую информацию в сжатом и точном изложении, то он обладает компетентностью "интеграция информации".
23/2006	Лекция 7. Описание познавательных деятельности, составляющих компетентность "создание информации". Школьник обладает компетентностью "создание информации", если он умеет сбалансированно осветить проблему на основе имеющейся, в том числе и противоречивой, информации. Контрольная работа № 2.
24/2006	Лекция 8. Описание познавательных деятельности, составляющих компетентность "сообщение информации". Могут ли ваши ученики адаптировать представленную информацию для конкретной аудитории, умеют ли грамотно цитировать источники, воздерживаться от провокационных высказываний при передаче информации конкретной аудитории?
Итоговая работа. Итоговая работа должна быть отправлена в Педуниверситет "Первое сентября" до 28 февраля 2007 г.	

Лекция 4.

Описание познавательных деятельности, составляющих компетентность "управление".

Примеры заданий на выработку этой компетентности

Всем знакома шутка, что людей беспокоят две проблемы: где взять еду и как похудеть. К сожалению, подобная ситуация возникает и в отношении информации. Как недостаток информации, так и ее избыток грозят человеку неприятностями в школе, на работе и, что хуже, неприятностями со здоровьем. Очень часто причиной всех этих неприятностей является недостаточное структурирование информации.

Вот что можно прочитать в Википедии относительно термина "структура". "Структура (от лат. *structūra* — строение) имеет целый спектр значений, встречающихся как в научной, так и в повседневной лексике. В своем основном значении структура есть *внутреннее устройство* чего-либо".

В "Словаре иностранных слов" относительно этого термина дается следующее определение: "Структура

(лат. *structūra*) — взаиморасположение и связь составных частей чего-либо; строение".

В определении ИКТ-компетентности когнитивная деятельность "управление" состоит из следующих действий (умений):

- создание схемы классификации для структурирования информации;
- использование предложенных классификационных схем для структурирования информации.

Исходя из определения деятельности "управление", можно сделать заключение, что вся деятельность по управлению информацией состоит в ее классификации. Однако это не совсем так. Данная трактовка термина "управление" привязана к определению ИКТ-компетентности [3], с которым мы работаем. Следует учитывать, что в соответствии с данным содержанием этого термина строятся и тестовые вопросы.

Далее разумно посмотреть, как термин “классификация” трактуется в энциклопедиях. Например, в Википедии можно прочитать: “*Классификация* — система группировки субъектов исследования или наблюдения в соответствии с их общими признаками”. “Словарь иностранных слов” дает следующее определение термина “классификация”: “*Классификация* (лат.) — распределение тех или иных объектов по классам (разделам, разрядам) в зависимости от их общих признаков”.

Прежде чем мы подробно рассмотрим когнитивную деятельность “управление”, отметим, что:

- структурирование информации невозможно без определения (или узнавания) информации;
- создание схемы классификации невозможно без анализа и выделения общих параметров-характеристик объектов рассматриваемого множества.

В тест по оценке ИКТ-компетентности девятиклассников, проводившийся в апреле—мае 2006 года в семи пилотных регионах, были включены три пятиминутных вопроса на умение структурировать информацию. В лекции 1 была приведена диаграмма с результатами тестирования по всем когнитивным умениям. Структурирование информации — умение, которым наши школьники владеют лучше всего: 63% девятиклассников получили высшую оценку. И это неудивительно — умению структурировать информацию учат и на уроках истории, и на уроках биологии, и на уроках информатики, вспомните, например, курс А.В. Горячева “Информатика в играх и задачах”. Однако 19% протестированных девятиклассников, т.е. каждый пятый, получили низшую оценку. Поэтому успокаиваться рано.

Классификация как структурирование

Очень часто информацию, содержащуюся в тексте, можно структурировать различными способами в зависимости от поставленной цели. Структурирование в смысле классификации подразумевает, что информация содержит описания группы объектов, которые надо соотнести друг с другом, сравнить и т.д.

Рассматривая классификацию как структурирование информации, можно выделить несколько вариантов классификации:

- параллельная классификация — деление объектов на непересекающиеся равнозначные множества;
- многоуровневая иерархия — деление объектов на непересекающиеся подчиненные множества, например, компьютерные файловые системы с каталогами, подкаталогами и т.д.;
- множественная классификация — деление объектов на пересекающиеся множества.

Предположим, учащимся требуется структурировать информацию в соответствии с предложенным вариантом классификации. Тогда учащиеся должны (1) воспользоваться описанием параметров классификации, если таковые есть, или (2) самостоятельно выделить (описать) признаки-параметры, которым должен удовлетворять объект для того, чтобы его можно было отнести

к конкретному множеству. В первом случае для успешного выполнения задания от учащихся требуется владение навыком “определения” (идентификации) информации. Во втором, более сложном случае, кроме навыка определения, учащийся должен владеть еще и навыком формального анализа, т.е. он должен уметь выделять и описывать для каждого объекта наиболее характерные признаки. Сложность состоит в том, что признаков, определяющих объект, как правило, бывает несколько, и надо выделить общие признаки для группы объектов.

Перечисляемые умения относятся к общеучебным, они формируются на протяжении всего периода обучения в школе. Однако следует обратить внимание на то, что в первую очередь формирование и развитие навыка структурирования текста через классификацию связано с навыком грамотной работы с текстом. Формирование у школьников ИКТ-компетентности на уроках требует от учителей использования специальных методов и приемов. Очевидно, что учитель должен быть настроен на формирование этой компетентности, т.е. помнить о данной задаче всегда. От учителя потребуются изменение дидактических целей типовых заданий, которые он дает своим учащимся (целей будет как минимум две: освоение конкретного учебного материала и формирование ИКТ-компетентности). Но успешность формирования ИКТ-компетентности будет зависеть и от методов и приемов, которые учитель использует в своей практике. На первое место среди “болевых” точек в педагогических методиках выходит владение учителем методики обучения функциональному чтению.

Рекомендации по работе с текстом

В нашей стране уже около десяти лет успешно работает общественная организация “Русская ассоциация чтения” (РАЧ), президентом которой является кандидат психологических наук, профессор Наталья Николаевна Сметанникова¹. РАЧ объединяет специалистов в разных областях знаний: педагогов и библиотекарей, психологов и социологов чтения, журналистов, студентов и родителей, ставящих своей целью продвижение чтения в России. В 1967 году ЮНЕСКО объявил день 8 сентября Международным днем распространения чтения и грамотности. “Сегодня быть грамотным — это иметь образование на уровне средней школы, уметь работать на компьютере, уметь общаться на родном и иностранном языках, уметь быстро и качественно работать с информацией. Всемирная организация здравоохранения включает показатель грамотности в 12 наиважнейших показателей, определяющих здоровье нации. ООН считает грамотность и продолжительность жизни одинаково важными характерис-

¹ “Русская ассоциация чтения” имеет отделения в Москве, Санкт-Петербурге, Екатеринбурге и Новороссийске. Вице-президенты РАЧ: профессор Б.В. Бирюкова, профессор Е.А. Гончарова, доцент В.П. Чудинова, О.А. Борисова, Е.Н. Овчинникова. Секретарь РАЧ — Д.В. Дмитриева. Контакты: тел. (495) 790-69-25, e-mail: rusredorg@gmail.com, www.rusreadorg.ru.

тиками жизни народа. Международный валютный фонд подсчитал, что экономическое развитие страны начинается тогда, когда уровень грамотности населения превышает 40%” [1].

Одним из направлений деятельности, в котором работают специалисты РАЧ, является разработка инновационных технологий и стратегий обучения чтению². Специалисты по чтению рекомендуют разные стратегии работы с текстом в зависимости от типа текста: научный, учебный, художественный, рекламный и т.д. [2]. К общим рекомендациям по работе с учебным текстом можно отнести следующее.

1. Работа с текстом начинается с его “сканирования”. Одна из задач “сканирования” — просмотреть текст и мысленно зафиксировать те места в тексте, в которых содержится информация, необходимая для ответа на вопросы задания. Фиксирование мест в тексте на бумажных носителях часто привязывается к физическому расположению отрывка текста относительно листа бумаги (внизу, сверху, после рисунка и т.д.). При работе с электронным текстом фиксировать такие места несколько сложнее, особенно если текст представляет собой непрерывный поток слов (без выделений, картинок, диаграмм и т.п.). При работе с источниками большого объема сканирование позволяет выделить фрагменты текста, требующие наиболее внимательного прочтения.

2. Далее ученик должен прочитать текст второй раз более внимательно, выделить (найти, сформулировать) ответы на поставленные вопросы.

Кроме того, специалисты РАЧ при работе с учебным текстом в классе советуют учителям употреблять словосочетание “работа с текстом” вместо “чтение текста”: это один из косвенных приемов, помогающих настроить учащихся на действительно продуманную работу.

Примеры учебных заданий на управление (классификацию)

1. Задание “История вычислительной техники”

Общее замечание к заданию. В школьном курсе информатики в 7–9-х классах изучается тема “История развития вычислительной техники”. Предлагаемое задание по этой теме содержит малоизвестные сведения: об ученых-исследователях Цузе и Атанасове школьники практически ничего не знают. Поэтому данная информация им будет интересна, тем более что текст содержит бытовые и исторические факты. Задание направлено на выработку умения выделять основные характеристики объекта, в частности, ЭВМ в соответствии с заданной классификацией.

Каждому учащемуся выдается лист с заданием или, если есть возможность работать в компьютерном классе, текст должен находиться в файле, например, формата *doc*.

² Заметим, что под чтением понимается не только и не столько собственно чтение, сколько функциональное чтение.

Конрад Цузе³

Создателем первого действующего компьютера с программным управлением считают немецкого инженера **Конрада Цузе** (1910–1995), который с детства любил изобретать и, еще когда учился в школе, сконструировал модель машины для размена денег.

О машине, способной выполнять вместо человека утомительные вычисления, он стал мечтать, будучи еще студентом. Не зная о работе Чарльза Бэббиджа, Цузе вскоре приступил к созданию устройства, во многом подобного Аналитической машине английского математика. В 1936 году, чтобы отдавать больше времени постройке компьютера, Цузе уволился из фирмы, где работал. На маленьком столе в доме родителей он устроил “мастерскую”. Примерно через два года компьютер, который занимал уже площадь около 4 м² и представлял собой хитросплетение реле и проводов, был готов.

Машина, названная им Z1 (от *Zuse* — фамилия Цузе, написанная по-немецки), имела клавиатуру для ввода данных. Результат вычислений появлялся на панели, для чего использовалось множество маленьких лампочек. В целом Цузе был доволен аппаратом, но считал ввод с клавиатуры неудобным и медленным, и спустя какое-то время появилась машина Z2, работавшая с перфоленкой. В 1941 году Цузе построил релейный компьютер Z3, который также использовал двоичную систему счисления. К сожалению, эти образцы машин были уничтожены при бомбардировках во время войны.

Джон Винсент Атанасов

В 1973 году через суд было установлено, что патентные права на основные идеи цифровых электронных машин принадлежат Джону Атанасову. Болгарин по происхождению, **Джон Винсент Атанасов** (1903–1995) стал американцем во втором поколении. Поиски путей к автоматизации вычислений Атанасов начал в 1933 году, когда руководил аспирантами, занимавшимися теорией упругости, квантовой физикой и физикой кристаллов. В большинстве задач, с которыми приходилось сталкиваться, фигурировали дифференциальные уравнения в частных производных. Для их решения приходилось использовать приближенные методы решения, они же, в свою очередь, требовали решения больших систем алгебраических уравнений. Ученый стал предпринимать попытки применения технических средств для ускорения вычислений: Атанасов задумал сконструировать компьютер, основанный на новых принципах, взяв при этом в качестве элементной базы электронные лампы.

Осенью 1939 года Джон Атанасов и его ассистент Клиффорд Берри приступили к постройке машины — специализированной ЭВМ, предназначенной для решения алгебраических уравнений с 30 неизвестными. Было решено назвать ее ABC (*Atanasoff Berry Computer*).

³ Текст взят из брошюры “Информатика в лицах: исторические зарисовки к урокам информатики”. М.: Чистые пруды, 2005.

Исходные данные, представленные в десятичной системе счисления, должны были вводиться в машину с помощью специальных перфокарт. Затем в самой машине осуществлялось преобразование десятичного кода в двоичный, который далее в ней и использовался.

К весне 1942 года работу над машиной удалось в основном завершить; однако в это время США уже находились в состоянии войны с нацистской Германией, и проблемы военного времени отодвинули работу над первой ЭВМ на задний план. Вскоре машину демонтировали.

Задание. Прочитайте текст о двух выдающихся ученых, создателях первых вычислительных машин, и определите, к какому поколению ЭВМ относятся вычислительные устройства Z1, Z3, ABC. Для этого

1) заполните табл. 1: если вычислительное устройство обладает указанной характеристикой, то в соответствующей клетке поставьте (+), нет — (–), ответа в тексте нет — (?);

2) проанализируйте таблицу и на основе характеристик, определяющих поколение ЭВМ, обоснуйте свой ответ.

Таблица 1

	Z1	Z3	ABC
Год создания			
Для ввода информации использовалась			
двоичная система счисления			
десятичная система счисления			
Для обработки информации в ЭВМ использовалась			
двоичная система счисления			
десятичная система счисления			
Аппаратная база			
механические реле			
электронные лампы			
Устройство ввода			
клавиатура			
считыватель с перфомент			
считыватель с перфокарт			
Устройство вывода			
панель с лампочками			
вывод на перфоменту			
вывод на перфокарты			

Ответ к заданию. Можно утверждать, что к первому поколению ЭВМ эти машины не относятся. Деление (классификация) ЭВМ по поколениям выполняется по типу элементной базы и по структуре. Вычислительные машины Z1 и Z3 были чисто механическими, а ЭВМ, относящиеся к I поколению, строились на основе электронно-вакуумных ламп. Вычислительная машина ABC была специализированной ЭВМ, поэтому, к сожалению, в данную классификацию мы ее включать не должны.

Правильное заполнение таблицы:

	Z1	Z3	ABC
Год создания	1938	1941	1942
Для ввода информации использовалась			
двоичная система счисления	–	–	–
десятичная система счисления	+/?	+/?	+
Для обработки информации в ЭВМ использовалась			
двоичная система счисления	+	+	+
десятичная система счисления	–	–	–
Аппаратная база			
механические реле	+	+	–
электронные лампы	–	–	+
Устройство ввода			
клавиатура	+	–	–
считыватель с перфомент	–	+	–
считыватель с перфокарт	–	–	+
Устройство вывода			
панель с лампочками	+	–	–
вывод на перфоменту	–	+	–
вывод на перфокарты	–	–	+

Замечание 1. Для выполнения задания предложите учащимся прочитать текст два раза (сканирование и пристальное чтение).

Замечание 2. При выполнении этого задания учащиеся должны использовать умение “определения” информации, в частности, находить в тексте информацию, заданную в неявном виде. Например, год создания Z1 явно в тексте не указан, его надо вычислить. В явном виде не указано, какая система счисления использовалась для ввода информации в Z1, поэтому ответ “?” допустим. Для установления этой характеристики учащиеся могут вспомнить, что Аналитическая машина Бэббиджа работала в десятичной системе счисления, но это очень сложное умозаключение. Также в явном виде для Z1 не указана внутренняя система счисления, эта характеристика устанавливается только на сопоставлении с машиной Z3.

Замечание 3. Для ответа на поставленный вопрос (для классификационного деления) учащиеся должны вспомнить, по каким признакам происходит деление ЭВМ на поколения, это, во-первых. А во-вторых, они должны четко понимать, какие объекты можно классифицировать по поколениям ЭВМ (для этого объекты должны быть универсальными вычислительными машинами).

Замечание 4. Сложность задания станет выше, если школьникам будет предложено самим составить таблицу характеристик вычислительных машин. Задание может быть сформулировано в таком виде: “Прочитайте текст о двух выдающихся ученых, создателях первых вычислительных машин. Составьте таблицу, в которой столбцами будут названия вычислительных устройств, о которых рассказывается в тексте, а строками — характеристики этих машин. Заполните сделанную вами таблицу. На основе анализа таблицы дайте ответ, к какому

поколению ЭВМ относятся перечисленные вычислительные устройства”.

Для успешного выполнения этого задания учащийся должен выполнить следующие действия:

- определить, о каких вычислительных машинах рассказывается в тексте;
- определить, о каких существенных для сравнения характеристиках рассказывается в тексте. Например, хотя характеристика “размер” (4 м² для Z1) сама по себе важна, она не включена в таблицу, т.к. мы ничего не знаем о размерах остальных машин;
- составить таблицу в соответствии с выбранными характеристиками (параметрами), т.е. дать названия строкам и столбцам;
- заполнить таблицу;
- сделать вывод по основному вопросу задания.

Замечание 5. Задание построено в необычной для российской школы форме. Задания такого вида учащимся давать полезно, но их обязательно надо обсуждать в классе. Как правило, школьники при ответе на вопрос: “К какому поколению относятся данные вычислительные устройства?” — концентрируют свое внимание на слове “каком”. И, соответственно, пытаются подобрать и обосновать ответ, исходя из того, что всего известно 5 поколений ЭВМ. Критичного отношения к вопросам мы у них не воспитываем. Поэтому они даже не допускают мысли, что вопрос-то “заковыристый”.

Замечание 6. Для некоторых школьников задание может оказаться “пустым”. Если они четко знают критерии классификации ЭВМ по поколениям, то сразу выделяют тип элементной базы. Но далеко не все девятиклассники смогут это сделать. А постепенное решение задания (через заполнение таблицы) формирует общий подход к решению задач на классификацию.

2. Пример тестового задания “Квартирный переезд”

Время выполнения задания — 5 минут. Проверяется умение структурировать информацию по предложенной схеме классификации.

Экзаменуемый должен:

- прочитать сценарий и задание;
- прочитать список книг и структурировать его в соответствии с предложенной схемой классификации;
- перетащить выбранные позиции списка в папки согласно предложенной схеме классификации.

Полный сценарий. Ваша семья собирается переезжать на новую квартиру. Дома не пройти, везде горы коробок. Вам досталось упаковывать семейную библиотеку. Чтобы проще было разбирать вещи на новой квартире, родители попросили вас разложить книги и периодики в коробки по направлениям (тематике). Вам нужно поместить в 1-ю коробку художественную литературу, во 2-ю коробку — учебники и справочную литературу, а в 3-ю коробку — периодическую печать.

После прочтения задания школьник нажимает кнопку “Далее” и переходит к следующему экрану, на котором он видит краткий сценарий и рабочее поле.

Краткий сценарий

1. Прочитайте список книг вашей библиотеки.
2. Определите, к какому направлению (тематике) относится каждая из них.
3. Перетащите (перетяните) выбранные вами позиции (книги) из списка в соответствующие папки (коробки).

Вам нужно поместить в 1-ю папку (коробку) художественную литературу, во 2-ю папку (коробку) — учебники и справочную литературу, а в 3-ю папку (коробку) — периодическую печать.

Рабочее поле

Библиотека	М.Стразнишас	Photoshop 5.5 для подготовки web-графики
Художественная литература	А.Блок	И невозможное возможно
Учебники и справочная литература	Наука и жизнь	№ 3 за 2002 год, № 2—4 за 2004 год, № 1—3 за 2005 год
Периодическая печать	Ф.М. Достоевский	Преступление и наказание
	А.А. Рывкин, А.З. Рывкин, Л.С. Хренов	Справочник по математике
	Д.Э. Розенталя, М.А. Теленкова	Словарь трудностей русского языка
	Тода Стауффер	Ваш компьютер Макинтош
	Туве Янсон	Все о муми-троллях
	PC WEEK	Подборка газет за последние 3 месяца
	А.Н. Толстой	Хождение по мукам
	П.А. Гуров	Русско-немецкий разговорник
	А.Д. Гетманова	Учебник по логике
	Книга для чтения по истории Средних веков	Пособие для учащихся средней школы
	И.С. Тургенев	Повести
	Навигатор игрового мира	№ 7 и 8 за 2005 год
	Литературная газета	№ 44—46 за 2005 год
Н.И. Сазонов (составитель)	Хрестоматия по русской и зарубежной литературе для 1—4-х классов	
Марина Цветаева	Стихи	
П.П. Ершов	Конек-горбунок	
Вокруг света	Подборка за 2004 год	
Э.Хемингуэй	Старик и море	

Пояснение

- При нажатии мышкой на любую часть из списка вся строка выделяется другим цветом.
- После перетаскивания позиции из списка в папку она удаляется из списка.

Ответы к заданию

Таблица 2

1-я папка “Художественная литература”	2-я папка “Учебная и справочная литература”	3-я папка “Периодическая печать”
А.Блок “И невозможное возможно”	М.Стразнижас “Photoshop 5.5 для подготовки web-графики”	“Наука и жизнь”, № 3 за 2002 год, № 2–4 за 2004 год, № 1–3 за 2005 год
Ф.М. Достоевский “Преступление и наказание”	А.А. Рывкин, А.З. Рывкин, А.С. Хренов “Справочник по математике”	“РС WEEK”, подборка газет за последние 3 месяца
Туве Янсон “Все о муми-троллях”	Д.Э. Розенталь, М.А. Теленкова “Словарь трудностей русского языка”	“Навигатор игрового мира”, № 7 и 8 за 2005 год
А.Н. Толстой “Хождение по мукам”	Тодд Стауффер “Ваш компьютер Макинтош”	“Литературная газета”, № 44–46 за 2005 год
И.С. Тургенев “Повести”	П.А. Гуров “Русско-немецкий разговорник”	“Вокруг света”, подборка за 2004 год
Марина Цветаева “Стихи”	А.Д. Гетманова “Учебник по логике”	
П.П. Ершов “Конек-горбунок”	“Книга для чтения по истории Средних веков”, пособие для учащихся средней школы	
Э.Хемингуэй “Старик и море”	Н.И. Сазонов (составитель) “Хрестоматия по русской и зарубежной литературе для 1–4-х классов”	

Таблица оценки выполнения задания “Квартирный переезд”

Компетентность	Уровень компетентности	Результат выполненных действий
Управление	Высокий	Заполняет папки в соответствии с таблицей № 2
	Средний	Заполняет папки в соответствии с таблицей № 2. Допускается одно несовпадение по любой позиции в таблице № 2
	Низкий	Заполняет папки с двумя или более ошибками

Замечание. Как ни странно, при выполнении этого задания наиболее часто возникающий вопрос у школьников был следующий: “А что такое периодическая печать?”

Вопросы и задания

1. По каким еще параметрам (характеристикам) можно структурировать информацию в тексте о создателях первых ЭВМ? Выделите как минимум три характеристики. Являются ли выделенные вами характеристики общими для любых компьютеров?

2. Опишите параметры классификации объектов из тестового задания “Квартирный переезд”.

Литература

1. 8 сентября. Международный день распространения чтения и грамотности. М.: Материалы Русской ассоциации чтения, 2006.

2. *Сметанникова Н.Н.* Стратегический подход к обучению чтению (междисциплинарные проблемы чтения и грамотности). М.: Школьная библиотека, 2005.

3. Отчет “ИКТ-компетентность в мировой практике. Показатель ИКТ-компетентности учащихся и работников образования как индикатор результативности Проекта ИСО”. М.: НФПК, 2005.

4. *Гейзов Г.* Рациональные приемы работы с книгой. М.: Книга, 1975.

5. *Колесова А.М.* Научная организация труда учащихся. Л.: Знание, 1975.

Вниманию слушателей курсов повышения квалификации!

На сайте “Педагогического университета” <http://edu.1september.ru> на странице каждого курса есть раздел “Консультации”. В нем вы можете задавать вопросы преподавателям и получать оперативные ответы.

Также на сайте размещаются учебные материалы в электронном виде (как правило, раньше доставки бумажных версий).

В “Личном кабинете” слушатели могут контролировать прохождение всех своих документов (отправку учебных материалов, получение и проверку контрольных работ и т.д.).

В текущем учебном году функциональность сайта была существенно расширена. Пользуйтесь новыми возможностями!

Уважаемые слушатели курсов повышения квалификации!

Для получения зачета по пройденной части курса вам необходимо выполнить контрольную работу № 1. Работа должна быть отправлена по адресу: ул. Киевская, д. 24, Москва, 121165, Педагогический университет “Первое сентября” не позднее 25 ноября 2006 г. Рекомендуем использовать полученные вами почтовые этикетки.

Вместе с выполненной работой необходимо выслать заполненный бланк (см. ниже) или его ксерокопию. Если у вас есть вопросы или замечания по контрольной работе или курсу в целом, пожалуйста, запишите их в поле “Комментарии”. Пожалуйста, сохраните у себя ксерокопию выполненной работы, чтобы исключить ее утрату при потере почтовых отправлений (при использовании бумажной почты такое редко, но случается).

Контрольная работа основана на материалах четырех первых лекций курса и содержит три задания. Оценивание работы будет производиться по системе “зачет/незачет”. После каждого задания указано максимальное количество баллов за его выполнение. Для того чтобы работа была зачтена, необходимо набрать не менее 8 баллов.

Контрольная работа № 1 по курсу И.Н. Фалиной, В.Ф. Бурмакиной “Как готовиться к тестированию по проверке ИКТ-компетентности школьников”

1. Объясните, что в современной педагогике понимается под функциональным чтением. Покажите роль и место функционального чтения в формировании ИКТ-компетентности.

(3 балла)

2. Приведите пример параграфа из школьного учебника информатики (7–9-е классы), прочитав который учащиеся могут сами составить схему или таблицу структурирования информации, содержащейся в тексте. Если позволяет учебный план, то предложите учащимся ва-

ших классов выполнить такое задание и опишите полученные результаты.

(4 балла)

3. Составьте одно или два задания на развитие умений “определение”, “доступ” или “управление”. В задании укажите, на формирование какого именно умения в основном оно нацелено, опишите форму проведения задания и способ его оценивания.

(5 баллов за каждое составленное задание)

Информатика

07-007

В.Ф. БУРМАКИНА, И.Н. ФАЛИНА

Как готовиться к тестированию по проверке ИКТ-компетентности школьников

Фамилия:

Имя:

Отчество:

Идентификатор: (указан в вашей персональной карточке)

Комментарии



САМАЯ ГОРЯЧАЯ ТЕМА

сегодняшнего российского образования
во всех газетах Издательского дома «Первое сентября»

в январе 2007 года вы получите

ИНФОРМАТИК 

специальный тематический выпуск
«ПРОФИЛЬНОЕ ОБУЧЕНИЕ»

В этом номере:

- Ликбез по теме «Информатика и профильная школа»
- Стандарт профильного уровня по информатике
- Результаты седьмого тура методического конкурса «Как это делаю я» по теме «Информатика в профиль». Практический опыт профильного обучения информатике
- Обзор изданных элективных курсов по информатике
- Примерные ответы на профильные билеты (продолжение осенних публикаций)

Шеф-редактор номера А.Л. Семенов, ректор МИОО, д.ф.-м.н., член-корр. РАО

Хотите быть в курсе?
Еще есть время подписаться
на газету «ИНФОРМАТИКА»

Подписной индекс 33371 по каталогу
агентства «Роспечать» «Газеты. Журналы»



При подписке на 6 месяцев — **подарок:** три выпуска
Библиотечки «Первое сентября» серии «Информатика»

Примерные ответы на профильные билеты

Е.А. ЕРЕМИН, А.П. ШЕСТАКОВ,

г. Пермь

Продолжение. Начало см. в № 19/2006

Билет № 3

1. Подходы к измерению информации. Преимущества и недостатки вероятностного и алфавитного подходов к измерению информации. Единицы измерения информации. Скорость передачи информации. Пропускная способность канала связи

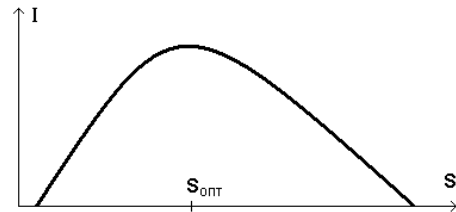
Информация является важнейшим понятием и основным объектом изучения в информатике (см. билет № 1). Неудивительно поэтому, что проблема измерения информации имеет фундаментальное значение¹.

На бытовом уровне информация является синонимом слов *сведения, знания, данные, новость, известие, сообщение* и аналогичным им. При этом неявно подразумевается, что тот, кто получает информацию, выделяет из нее некоторый **СМЫСЛ** (содержание).

Смысловая составляющая информации во многом индивидуальна. Большинство россиян не способны извлечь никакой информации из текста на японском языке. Многие взрослые, взяв учебник для начальных классов, также не сочтут его заслуживающей своего внимания информацией, хотя, в отличие от предыдущего случая, понимают (слишком хорошо!), что там написано. Химика редко интересуют сообщения об археологических открытиях, а большая часть литераторов активно игнорирует любые сведения из области математики. Наконец, многие образованные люди не верят в статьи, опубликованные в бульварной прессе, заранее считая их недостоверными. Таким образом, информативность любых сведений и сообщений существенно зависит от воспринимающего их человека, его предыдущих знаний, опыта, интересов, отношения к источнику информации и множества других факторов личного характера, т.е. по своей сути является **субъективной**.

Дополнительное пояснение. Во всех наиболее распространенных школьных учебниках подобными описаниями и несколькими несвязанными примерами неинформативных сообщений дело и ограничивается. Но на самом деле вопрос о соотношении между информативностью и предыдущими знаниями человека имеет самостоятельное значение. Например, в недавно вышедших учебниках [1, 2] приведен следующий весьма на-

глядный и любопытный качественный график зависимости воспринимаемой пользователем информации I от предвзятительно известных ему сведений S (для обозначения совокупности сведений, которыми располагает пользователь или любая другая система, принят специальный научный термин — *тезаурус*).



Из графика отчетливо видно, что воспринимаемая из фиксированного сообщения информация зависит от познаний пользователя неоднозначно. При малых S (тезаурус мал, пользователь неквалифицирован) воспринимаемая информация близка к нулю — человек ее просто не понимает. По мере роста S ситуация улучшается, и количество воспринятой из данного сообщения информации растет (невежественный пользователь быстро прогрессирует). Но лишь до определенного предела, а затем величина I начинает постепенно уменьшаться. Здесь вступает в действие другой фактор: при больших S (большой объем знаний, мы имеем дело с образованным пользователем) многие сообщения не могут добавить к тезаурусу ничего нового и их информативность вновь устремляется к нулю. Выражаясь житейским языком, когда человек много знает, его ничем не удивишь.

Таким образом, график количества воспринимаемой информации имеет максимум (на рисунке он обозначен $S_{\text{опт}}$), соответствующий вполне определенному соотношению между содержащейся в сообщении информацией и уже имеющимися у субъекта знаниями².

Если задуматься над сутью рассмотренного графика, то можно получить целый ряд практически интересных выводов. Например, что передаваемая информация должна определенным образом соотноситься с уже имеющимися в ее приемнике сведениями, иначе данные могут оказаться бесполезными и только напрасно будут загружать каналы связи и узлы обработки. Между прочим, учащиеся после этих рассуждений должны проникнуться еще большим уважением к учителю, который постоянно вынужден “подстраивать” степень информативности своего рассказа, интуитивно оценивая разнообразные объемы тезаурусов сидящих в классе учеников, причем с учетом возможности их интеллектуального роста по мере освоения материала.

¹ Проблема существенна, если информатику считать происходящей от математики; если же, как это модно в последнее время, гипертрофированно выпячивать философскую сторону вопросов, тогда актуальность количественных измерений теряется, а расчетные задачи становятся и вовсе неинтересными (например, философов мало занимает, как именно и в каких единицах физики измеряют массу вещества или характеристики полей).

² На самом деле картина еще сложнее: при $S \rightarrow \infty$ график, видимо, должен приближаться к оси асимптотически (нельзя назвать **абсолютно** все), по мере приема сообщения тезаурус может расти и т.д.; но даже приведенная упрощенная картинка весьма наглядно объясняет многие особенности восприятия информации.

Несколько иная трактовка измерения смысла сообщений по способу А.Н. Колмогорова весьма просто и интересно обсуждается в школьном учебнике [3].

Таким образом, субъективный характер восприятия информации делает однозначное измерение количества информации весьма затруднительным. Заметим, что современным компьютерам смысл обрабатываемых данных вообще принципиально недоступен, что делает еще более призрачной надежду на решение проблемы автоматического измерения “количества” содержания, которое в этих данных заключено.

Как же все-таки измерить информацию? Общепринятым на данный момент решением проблемы является измерение объема информации при полном игнорировании ее смысла. Такой подход, несмотря на кажущуюся бессмысленность, оказывается необычайно полезным и широко применяется на практике.

Действительно, в целом ряде важных практических задач смысл информации и даже ее вид (числа, текст, видео) несущественен. Например, при передаче информации по каналам связи, при распределении объемов ОЗУ для хранения различных типов данных, при записи информации на внешние носители, при архивации и многих других компьютерных применениях содержание передаваемой и обрабатываемой информации особого значения не имеет.

Нечто похожее наблюдается и в “некомпьютерных” областях. Так, книги хранятся и ищутся не по содержанию, а по другим, часто весьма формальным признакам; в библиотеке нашего университета, в частности, книги расставлены на стеллажах в том числе с учетом размера обложки, который явно слабо связан с содержанием книги. Аналогично почтальону должно быть все равно, что именно находится внутри доставляемого им конверта, а диктор телевидения не может пропускать отдельные новости или их фрагменты в соответствии со своими личными убеждениями.

Примечание. Подобно тому, как в физике при полном игнорировании трения можно установить фундаментальные законы движения, можно надеяться, что изучение “информации без смысла” позволит понять наиболее важные закономерности протекания информационных процессов.

На этих и многочисленных подобных примерах мы видим, что информация перестает зависеть от человека при абстрагировании от ее смысла. Следовательно, появляется возможность **объективного** измерения количества информации. При этом используется два подхода: *вероятностный* или *алфавитный*.

Вероятностный подход к измерению информации

Любая информация может рассматриваться как уменьшение неопределенности наших знаний об окружающем мире (в теории информации принято говорить именно об уменьшении неопределенности, а не об увеличении объема знаний). Математически это высказывание эквивалентно простой формуле

$$I = H_1 - H_2$$

где I — это количество информации, а H_1 и H_2 — начальная и конечная неопределенность соответственно (очевидно, что $H_1 \geq H_2$). Величину H , которая описывает степень неопределенности, в литературе принято называть *энтропией*.

Важным частным случаем является ситуация, когда некоторое событие с несколькими возможными исходами уже произошло, а, значит, неопределенность его результата исчезла. Тогда $H_2 = 0$ и формула для информации упрощается:

$$I = H$$

Таким образом, энтропия опыта равна той информации, которую мы получаем в результате его осуществления. И наоборот: информация, получаемая из опыта, может быть вычислена через его энтропию. Очевидно, что единицы измерения информации и энтропии совпадают.

Вычисление энтропии при вероятностном подходе базируется на рассмотрении данных о результате некоторого случайного события, т.е. события, которое может иметь несколько исходов. Случайность события заключается в том, что реализация того или иного исхода имеет некоторую степень неопределенности.

Пусть, например, абсолютно незнакомый нам ученик сдает экзамен, результатом которого может служить получение оценок 2, 3, 4 или 5. Поскольку мы ничего не знаем о данном ученике, то степень неопределенности всех перечисленных результатов сдачи экзамена совершенно одинакова. Напротив, если нам известно, как он учится, то уверенность в некоторых исходах будет больше, чем в других: так, отличник скорее всего сдаст экзамен на пятерку, а получение двойки для него — это нечто почти невероятное.

Наиболее просто определить количество информации в случае, когда все исходы события могут реализоваться с равной долей вероятности. В этом случае для вычисления информации используется **формула Хартли**. В более сложной ситуации, когда исходы события ожидаются с разной степенью уверенности, требуются более сложные вычисления по **формуле Шеннона**, которую обычно выносят за рамки школьного курса информатики. Очевидно, что формула Хартли является некоторым частным случаем более общей формулы Шеннона.

Формула Хартли была предложена в 1928 году американским инженером Р.Хартли. Она связывает количество равновероятных состояний N с количеством информации I в сообщении о том, что любое из этих состояний реализовалось. Наиболее простая форма для данной формулы записывается следующим образом:

$$2^I = N$$

Причем обычно значение N известно, а I приходится подбирать, что не совсем удобно. Поэтому те, кто знает математику получше, предпочитают преобразовать данную формулу так, чтобы сразу выразить искомую величину I в явном виде:

$$I = \log_2 N$$

Важный частный случай получается из приведенной формулы при $N = 2$, когда результатом вычисления является единичное значение. Единица информации носит название бит (от англ. *Binary digiT* — двоичная цифра); таким образом, **1 бит — это информация о результате опыта с двумя равновероятными исходами**. Чем больше возможных исходов, тем больше информации в сообщении о реализации одного из них.



Пример 1. Из колоды выбрали 16 карт (все “картинки” и тузы) и положили на стол рисунком вниз. Верхнюю карту перевернули (см. рисунок). Сколько информации будет заключено в сообщении о том, какая именно карта оказалась сверху?

Все карты одинаковы, поэтому любая из них могла быть перевернута с одинаковой вероятностью. В таких условиях применима формула Хартли.

Событие, заключающееся в открытии верхней карты, для нашего случая могло иметь 16 возможных исходов. Следовательно, информация о реализации одного из них равняется

$$I = \log_2 16 = 4 \text{ бита}$$

Примечание. Если вы не любите логарифмы, можно записать формулу Хартли в виде $2^I = 16$ и получить ответ, подбирая такое I , которое ей удовлетворяет.

Пример 2. Решите предыдущую задачу для случая, когда сообщение об исходе случайного события было следующим: “верхняя перевернутая карта оказалась черной дамой”.

Отличие данной задачи от предыдущей заключается в том, что в результате сообщения об исходе случайного события не наступает полной определенности: выбранная карта может иметь одну из двух черных мастей. В этом случае, прежде чем воспользоваться формулой Хартли, необходимо вспомнить, что информация есть уменьшение неопределенности знаний:

$$I = H_1 - H_2$$

До переверота карты неопределенность (энтропия) составляла

$$H_1 = \log_2 N_1$$

после него —

$$H_2 = \log_2 N_2$$

(причем для нашей задачи $N_1 = 16$, а $N_2 = 2$).

В итоге информация вычисляется следующим образом:

$$I = H_1 - H_2 = \log_2 N_1 - \log_2 N_2 = \log_2 N_1 / N_2 = \log_2 16/2 = 3 \text{ бита}$$

Заметим, что в случае, когда нам называют карту точно (см. предыдущий пример), неопределенность результата исчезает, $N_2 = 1$, и мы получаем “традиционную” формулу Хартли. И еще одно полезное наблюдение. Полная информация о результате рассматриваемого опыта составляет 4 бита (см. пример 1). В данном же случае мы получили 3 бита информации, а оставшийся четвертый описывает сохранившуюся неопределенность выбора между двумя дамами черной масти.

Алфавитный (объемный) подход к измерению информации

Помимо описанного выше вероятностного подхода к измерению информации, состоящего в подсчете неопределенности исходов того или иного события, существует и другой. Его часто называют *объемным*, и он заключается в определении количества информации в каждом из знаков дискретного сообщения с последующим подсчетом количества этих знаков в сообщении.

Пусть сообщение кодируется с помощью некоторого набора знаков. Заметим, что если для данного набора установлен порядок следования знаков, то он называется *алфавитом*. Наиболее сложной частью работы при объемном измерении информации является определение количества информации, содержащейся в каждом отдельном символе: остальная часть процедуры весьма проста. Для определения информации в одном символе алфавита можно также использовать вероятностные методы, поскольку появление конкретного знака в конкретном месте текста есть явление случайное.

Самый простой метод подсчета заключается в следующем. Пусть алфавит, с помощью которого записываются все сообщения, состоит из M символов. Для простоты предположим, что все они появляются в тексте с одинаковой вероятностью (конечно, это грубая модель³, но зато очень простая). Тогда в рассматриваемой постановке применима формула Хартли для вычисления информации об одном из исходов события (о появлении любого символа алфавита):

$$I = \log_2 M$$

Поскольку все символы “равноправны”, естественно, что объем информации в каждом из них одинаков. Следовательно, остается полученное значение I умножить на количество символов в сообщении, и мы получим общий объем информации в нем. Напомним читателям, что осмысленность сообщения в описанной процедуре нигде не требуется, напротив, именно при отсутствии смысла предположение о равновероятном появлении всех символов выполняется лучше всего!

Примечание. Стоит знать, что описанный простой способ кодирования, когда коды всех символов имеют одинаковую длину, не является единственным. Часто при передаче или архивации информации по соображениям экономичности тем символам, которые встречаются чаще, ставятся в соответствие более короткие коды и наоборот. Мы здесь не будем рассматривать этот весьма интересный и практически важный вопрос. Желающие могут обратиться, например, к известному школьному учебнику информатики [4] (начиная со второго издания) или к более глубокому, но тоже достаточно понятному [5].

Можно показать, что при **любом варианте кодирования**

$$I_{\text{вероятностная}} \leq I_{\text{объемная}}$$

(чем экономичнее способ кодирования, тем меньше разница между этими величинами — см. пример 4, приведенный ниже).

³ В действительности в осмысленном тексте символы появляются с разной вероятностью; кроме того, не все сочетания букв в естественных языках встречаются одинаково часто.

Пример 3. Определить информацию, которую несет в себе 1-й символ в кодировках ASCII и Unicode.

В алфавите ASCII предусмотрено 256 различных символов, т.е. $M = 256$, а

$$I = \log_2 256 = 8 \text{ бит} = 1 \text{ байт}$$

В современной кодировке Unicode заложено гораздо большее количество символов. В ней определено 256 алфавитных страниц по 256 символов в каждой. Предполагая для простоты, что все символы используются, получим, что

$$I = \log_2 (256 * 256) = 8 + 8 = 16 \text{ бит} = 2 \text{ байта}$$



Пример 4. Текст, сохраненный в коде ASCII, состоит исключительно из арифметических примеров, которые записаны с помощью 10 цифр от 0 до 9, 4 знаков арифметических операций, знака равенства и некоторого служебного кода, разделяющего примеры между собой. Сравните количество информации, ко-

торое несет один символ такого текста, применяя вероятностный и алфавитный подходы.

Легко подсчитать, что всего рассматриваемый в задаче текст состоит из $N = 16$ различных символов. Следовательно, по формуле Хартли

$$I_{\text{вероятностная}} = \log_2 16 = 4 \text{ бита}$$

В то же время, согласно вычислениям примера 3, для символа ASCII

$$I_{\text{алфавитная}} = 8 \text{ бит}$$

Двукратный избыток при кодировании символов связан с тем, что далеко не все коды ASCII оказываются в нашем тексте востребованными. В то же время несложно построить вариант специализированной 4-битной кодировки для конкретной задачи⁴, для которого $I_{\text{вероятностная}}$ и $I_{\text{алфавитная}}$ окажутся равными.

В порядке подведения итогов сравним вероятностный и алфавитный подходы, как того требует вопрос билета. Первый подход позволяет вычислить предельное (минимально возможное) *теоретическое* значение количества информации, которое несет сообщение о данном исходе события. Второй — каково количество информации *на практике* с учетом конкретной выбранной кодировки. Очевидно, что первая величина есть однозначная характеристика рассматриваемого события, тогда как вторая зависит еще и от способа кодирования: в “идеальном” случае обе величины совпадают, однако на практике используемый метод кодирования может иметь ту или иную степень избыточности.

⁴ Достаточно каждому из 16 символов поставить в соответствие двоичное число от 0 до 15.

С рассмотренной точки зрения вероятностный подход имеет преимущество. Но, с другой стороны, алфавитный способ заметно проще и с некоторых позиций (например, для подсчета требуемого количества памяти) полезнее.

Примечание. В учебниках информатики обычно ограничиваются описанием обоих подходов и не производится их сравнение. Приведенное выше сопоставление авторы провели исходя из собственных представлений. Возможно, составители билетов имели в виду какие-либо еще преимущества и недостатки.

Вопрос о **единицах измерения** информации уже возникал при обсуждении вероятностного и алфавитного подходов. В самом деле, трудно изложить способ измерения величины, не упоминая при этом о единицах ее измерения. Поэтому мы уже сформулировали выше, что с теоретической точки зрения **1 бит — это информация, которая сокращает неопределенность наших знаний вдвое** (ответ на вопрос типа “да”/“нет”, наличие или отсутствие какого-либо свойства, четность числа и т.д.). С точки зрения практической реализации компьютерных устройств для обработки информации 1 бит — это отдельный двоичный разряд любого из таких устройств. Иначе говоря, в вычислительной технике бит служит конструктивной базой для построения всех цифровых двоичных устройств: регистров, сумматоров и т.п. Отсюда очевидно, что в теории информации количество бит может быть любым, в том числе дробным, в то время как в реальных устройствах оно обязательно целое.

Бит, будучи минимально возможной порцией информации в компьютере, довольно маленькая единица измерения. Поэтому на практике чаще всего используется другая единица, которая называется **1 байт** = 8 бит. С точки зрения устройства компьютера байт замечателен тем, что является минимальной **адресуемой** информацией в компьютере, иначе говоря, считать из памяти часть байта невозможно. В современных компьютерах все устройства памяти имеют байтовую структуру, а внешние устройства также обмениваются информацией байтами или кратными ему порциями. Как следствие все типы данных (числа, символы и др.) представляются в компьютере величинами, кратными байту.

Примечание. Даже логические переменные, для каждой из которых, казалось бы, достаточно 1 бита, обычно занимают в оперативной памяти полный байт (или иногда ради единообразия даже несколько байт, например, LongBool в Паскале).

С целью получения шкалы для измерения объемов информации в широких пределах от байта с помощью стандартных приставок образуется целая система более крупных производных единиц:

$$1 \text{ килобайт} = 1024 \text{ байта}$$

$$1 \text{ мегабайт} = 1024 \text{ килобайта}$$

$$1 \text{ гигабайт} = 1024 \text{ мегабайта}$$

и т.д. В отличие от общепринятой системы производных единиц (широко используемой, например, в физике) при пересчете применяется множитель 1024, а не 1000.

Причина заключается в двоичном характере представления информации в компьютере: $1024 = 2^{10}$, и, следовательно, лучше подходит к измерению двоичной информации.

Научившись измерять количество информации, можно ставить вопрос, как быстро она передается. Величину, которая равна количеству информации, передаваемому за единицу времени, принято называть *скоростью передачи информации*. Очевидно, что если за время t по каналу связи передано количество информации I , то скорость передачи вычисляется как отношение I / t .

Примечание. При практической работе с величиной скорости передачи информации следует очень внимательно относиться к тому, что именно понимается под передаваемой информацией I . В частности, в процессе передачи к собственно пользовательской информации может добавляться значительное количество служебных, вспомогательных данных: например, согласно сетевому протоколу UDP (*User Datagram Protocol*), который является некоторой разновидностью известного протокола TCP (*Transmission Control Protocol*), из 146 байт стандартного Ethernet-кадра 46 являются служебными [6]. Кроме того, непосредственно перед передачей данные могут сжиматься или шифроваться, что также повлияет на время их передачи.

Скорость передачи данных нельзя сделать сколь угодно большой; ее предельная максимальная величина имеет специальное название — *пропускная способность* канала связи. Данная характеристика определяется устройством канала и, что не так очевидно, способом передачи сигналов по нему. Иными словами, для разных способов представления данных одна и та же линия связи может иметь разную пропускную способность.

К.Шеннон в созданной им теории информации доказал, что достигнуть при передаче пропускной способности линии можно всегда и путем к этому является повышение эффективности кодирования. Более того, даже при наличии в канале шумов *любого* уровня *всегда* можно закодировать сообщение таким образом, чтобы не происходило потери информации [1, 5].

Обе величины — скорость передачи и пропускная способность — по определению измеряются в одних и тех же единицах, являющихся отношением единиц информации и времени: бит/с, байт/с, Кб/с и т.д.

Дополнительное пояснение. Кроме того, существует еще одна родственная единица измерения параметров передачи — *бод*. Количество бод есть количество изменений *информационного параметра* в секунду. Скорость передачи в битах в секунду в общем случае не совпадает с количеством бод. В [1] приводится очень наглядный пример, когда скорость в бит/с вдвое выше, чем число бод. “Если информационными параметрами являются фаза и амплитуда синусоиды, причем различают 4 состояния фазы (0, 90, 180 и 270) и два значения амплитуды, то информационный сигнал имеет восемь различимых состояний. В этом случае модем, работающий со скоростью 2400 бод (с тактовой частотой 2400 Гц), передает информацию со скоростью 7200 бит/с, так как при одном изменении сигнала передается три бита информации”. Возможно, кстати, и обратное соотношение между величинами в бит/с и бод; в частном случае они могут совпадать.

В качестве примера типичных значений скоростей передачи данных в современных компьютерах ниже приводятся табл. 1 и 2, составленные на основе сведений из известной книги [7].

Таблица 1. Характеристики устройств внешней памяти

Тип диска	Емкость	Скорость передачи данных (Мб/с)
FDD 5"	360 К	0,027
FDD 5"	1,2 М	0,045
FDD 3,5"	1,44 М	0,055
HDD IDE	10–30 Г	2–20
HDD SCSI	10–30 Г	2–40
CD-ROM 1x	650 М	0,15
CD-ROM 48x	650 М	7,2
DVD-ROM 12x	5–17 Г	130
Iomega Zip	100 М	1,4
Iomega Zip	250 М	2,4
Iomega Jaz	1 Г	7,5
Compact Flash	32 М	до 20

Таблица 2. Характеристики шин расширения

Шина	Пропускная способность (Мб/с)	Частота (МГц)
ISA-8	4	8
ISA-16	8	8
EISA	33,3	8,33
VLB	132	33–66
PCI	132/264	33/66

Примечание. Хотя проблема пропускной способности каналов связи весьма подробно излагается в специальной литературе, в доступных для учителей и школьников источниках она рассматривается не всегда, а если и рассматривается, то весьма поверхностно. Поэтому на экзамене, по мнению авторов, надо требовать от учеников знания только самых минимальных сведений. Расширенный материал в нашей публикации приведен исключительно для того, чтобы дать некоторую общую ориентировку учителям. Нам кажется, что это один из примеров, когда, прежде чем требовать знания вопроса от учащихся, стоит описать его на нужном уровне в школьных учебниках.

Литература

1. Акулов О.А., Медведев Н.В. Информатика: базовый курс. М.: Омега-Л, 2005, 552 с.
2. Бройдо Э.А., Ильина О.П. Архитектура ЭВМ и систем. СПб.: Питер, 2006, 718 с.
3. Информационная культура: Кодирование информации, информационные модели. 9–10-е классы. М.: Дрофа, 2000, 208 с.
4. *Семакин И.Г.* Информатика. Базовый курс. 7–9-е классы / И.Г. Семакин, Л.А. Залогова, С.В. Русаков, Л.В. Шестакова. 2-е изд. М.: БИНОМ, 2004, 390 с.
5. *Стариченко Б.Е.* Теоретические основы информатики. М.: Горячая линия — Телеком, 2003, 312 с.
6. *Никифоров С.В.* Введение в сетевые технологии. М.: Финансы и статистика, 2003, 224 с.
7. Аппаратные средства IBM PC. Энциклопедия. / М.Гук. СПб.: Питер, 2003, 923 с.

2. С использованием электронной таблицы произвести обработку данных с помощью статистических функций

Даны сведения об учащихся класса, включающие средний балл за четверть, возраст (год рождения) и пол. Определить средний балл мальчиков, долю отличниц среди девочек и разницу среднего балла учащихся разного возраста.

Решение

Заполним таблицу исходными данными и проведем необходимые расчеты. В таблицу будем заносить данные из школьного журнала.

Сведения об учащихся класса								
№	Фамилия	Имя	Ср. балл	Дата рожд.	Пол	Возраст	Отличница?	
1	Иванов	Вася	3	04.09.90	м	16	0	
2	Петрова	Оля	3,7	05.02.91	ж	15	0	
3	Сидорова	Оля	4,2	01.12.90	ж	15	0	
4	Васечкин	Коля	5	13.06.90	м	16	0	
5	Семенова	Аня	5	22.06.90	ж	16	1	
6	Аникина	Лена	4,9	01.01.91	ж	15	0	
7	Суворова	Ирина	4,5	07.04.90	ж	16	0	
8	Андреева	Лёля	4	05.05.90	ж	16	0	
9	Сумкин	Ваня	3,9	03.02.91	м	15	0	
10	Кузовлев	Рома	4,9	17.03.90	м	16	0	
11	Фомичев	Саша	4,2	22.08.90	м	16	0	
12	Арбузова	Саша	3,6	31.03.90	ж	16	0	
18	Средний балл мальчиков		4,20					
19	Доля отличниц среди девочек		0,14					
20	Разница ср. балла учащихся разного возраста		0,10					

В таблице используются дополнительные колонки, которые необходимы для ответа на вопросы, поставленные в задаче (текст в них записан синим цветом), — *возраст ученика* и является ли учащийся *отличником* и *девочкой* одновременно.

Для расчета возраста использована следующая формула (на примере ячейки G4):

$$=ЦЕЛОЕ((СЕГОДНЯ()-E4)/365,25)$$

Прокомментируем ее. Из сегодняшней даты вычитается дата рождения ученика. Таким образом, получаем

полное число дней, прошедших с рождения ученика. Разделив это количество на 365,25 (реальное количество дней в году, 0,25 дня для обычного года компенсируется високосным годом), получаем полное количество лет ученика; наконец, выделив целую часть, — возраст ученика.

Является ли девочка отличницей, определяется формулой (на примере ячейки H4):

$$=ЕСЛИ(И(D4=5;F4="ж");1;0)$$

Приступим к основным расчетам.

Прежде всего требуется определить средний балл мальчиков. Согласно определению, необходимо разделить суммарный балл мальчиков на их количество. Для этих целей можно воспользоваться соответствующими функциями табличного процессора.

$$=СУММЕСЛИ(F4:F15;"м";D4:D15)/СЧЁТЕСЛИ(F4:F15;"м")$$

Функция СУММЕСЛИ позволяет просуммировать значения только в тех ячейках диапазона, которые отвечают заданному критерию (в нашем случае ребенок является мальчиком). Функция СЧЁТЕСЛИ подсчитывает количество значений, удовлетворяющих заданному критерию. Таким образом и получаем требуемое.

Для подсчета доли отличниц среди всех девочек отнесем количество девочек-отличниц к общему количеству девочек (здесь и воспользуемся набором значений из одной из вспомогательных колонок):

$$=СУММ(H4:H15)/СЧЁТЕСЛИ(F4:F15;"ж")$$

Наконец, определим отличие средних баллов разного возрастных детей (воспользуемся в расчетах вспомогательной колонкой *Возраст*):

$$=ABS(СУММЕСЛИ(G4:G15;15;D4:D15)/СЧЁТЕСЛИ(G4:G15;15)-СУММЕСЛИ(G4:G15;16;D4:D15)/СЧЁТЕСЛИ(G4:G15;16))$$

Таким образом, задача полностью решена. На рисунке представлены результаты решения для заданного набора данных.

Варианты заданий

С использованием электронной таблицы произвести обработку данных с помощью статистических функций.

1. Даны сведения об учащихся класса, включающие оценки в течение одного месяца. Подсчитайте количество пятерок, четверок, двоек и троек, найдите средний балл каждого ученика и средний балл всей группы. Создайте диаграмму, иллюстрирующую процентное соотношение оценок в группе.

2. Четверо друзей путешествуют на трех видах транспорта: поезде, самолете и пароходе. Николай проплыл 150 км на пароходе, проехал 140 км на поезде и пролетел 1100 км на самолете. Василий проплыл на пароходе 200 км, проехал на поезде 220 км и пролетел на самолете 1160 км. Анатолий пролетел на самолете 1200 км, проехал поездом 110 км и проплыл на пароходе 125 км. Мария проехала на поезде 130 км, пролетела на самолете 1500 км и проплыла на пароходе 160 км.

Построить на основе вышеперечисленных данных электронную таблицу.

- Добавить к таблице столбец, в котором будет отобразиться общее количество километров, которое проехал каждый из ребят.

- Вычислить общее количество километров, которое ребята проехали на поезде, пролетели на самолете и проплыли на пароходе (на каждом виде транспорта по отдельности).

- Вычислить суммарное количество километров всех друзей.

- Определить максимальное и минимальное количество километров, пройденных друзьями по всем видам транспорта.

- Определить среднее количество километров по всем видам транспорта.

3. Создайте таблицу “Озера Европы”, используя следующие данные по площади (кв. км) и наибольшей глубине (м): Ладожское 17 700 и 225; Онежское 9510 и 110; Каспийское море 371 000 и 995; Венерн 5550 и 100; Чудское с Псковским 3560 и 14; Балатон 591 и 11; Женевское 581 и 310; Веттерн 1900 и 119; Боденское 538 и 252; Меларен 1140 и 64. Определите самое большое и самое маленькое по площади озеро, самое глубокое и самое мелкое озеро.

4. Создайте таблицу “Реки Европы”, используя следующие данные длины (км) и площади бассейна (тыс. кв. км): Волга 3688 и 1350; Дунай 2850 и 817; Рейн 1330 и 224; Эльба 1150 и 148; Висла 1090 и 198; Луара 1020 и 120; Урал 2530 и 220; Дон 1870 и 422; Сена 780 и 79; Темза 340 и 15. Определите самую длинную и самую короткую реку, подсчитайте суммарную площадь бассейнов рек, среднюю протяженность рек европейской части России.

5. В банке производится учет своевременности выплат кредитов, выданных нескольким организациям. Известна сумма кредита и сумма, уже выплаченная организацией. Для должников установлены штрафные санкции: если фирма выплатила кредит более чем на 70 процентов, то штраф составит 10 процентов от суммы задолженности, в противном случае штраф составит 15 процентов. Посчитать штраф для каждой организации, средний штраф, общее количество денег, которые банк собирается получить дополнительно. Определить средний штраф бюджетных организаций.

3. Решить текстовую логическую задачу (необходимо использовать не менее четырех переменных)

Решить текстовую логическую задачу:

Болельщики футбольных команд делали прогнозы об итогах соревнований “Турнир четырех”:

— Я уверен, что “Спартак” будет чемпионом, а “ЦСКА” займет последнее место, — сказал Иван.

— Что ты, “Спартак” выше третьего не поднимется, а “ЦСКА” станет вторым, — возразил Сергей.

— Чемпионом будет “Динамо”, а “ЦСКА” войдет в тройку сильнейших, — сделал свой прогноз Петр.

— “Динамо” будет вторым, а вот “Ротор” точно будет последним, — промолвил Алексей.

Выяснилось, что каждый из болельщиков был прав в одном прогнозе и ошибся во втором. Как распределены места, занятые командами?

Решение

Решим задачу путем сопоставления высказываний и опираясь на информацию о том, что одно из них истинно, другое — ложно.

Анализ начнем с последнего высказывания.

Предположим, что высказывание “Ротор” займет 4-е место” истинно. Тогда “Динамо” будет вторым” — ложно (т.е. “Динамо” занимает 1-е или 3-е место).

Пусть истинно высказывание “Динамо” займет первое место”, тогда ложно высказывание “ЦСКА” войдет в тройку сильнейших”, т.е. “ЦСКА” занимает 4-е место, но это место у нас уже занял “Ротор”. Поэтому эта цепочка рассуждений неверна. Следовательно, истинным будет высказывание “ЦСКА” займет 2-е или 3-е место”, а ложным — “Динамо” займет 1-е место”. Для “Динамо” осталась единственная возможность — 3-е место.

Тогда далее высказывание “ЦСКА” станет вторым” истинно, а “Спартак” выше третьего не поднимется” — ложно. Таким образом, для “Спартака” остается единственная возможность — 1-е место, что не противоречит высказываниям Ивана, исходя из того, что одно из них истинно, другое — ложно.

Таким образом, распределение мест: “Спартак” — I, “ЦСКА” — II, “Динамо” — III, “Ротор” — IV.

Если отталкиваться от посыла, что истинно высказывание “Динамо” будет вторым”, ложно “Ротор” будет четвертым”, приходим к противоречию (проделать рассуждения самостоятельно”).

Варианты заданий⁵

1. Три подразделения — А, В, С — торговой фирмы стремились получить по итогам года максимальную прибыль. Экономисты высказали следующие предположения:

1) А получит максимальную прибыль только тогда, когда получат максимальную прибыль В и С,

2) Либо А и С получают максимальную прибыль одновременно, либо одновременно не получают,

3) Для того чтобы подразделение С получило максимальную прибыль, необходимо, чтобы и В получило максимальную прибыль.

По завершении года оказалось, что одно из трех предположений ложно, а остальные два истинны. Какие из названных подразделений получили максимальную прибыль?

2. Задача “Валютные махинации”.

В нарушении правил обмена валюты подозреваются четыре работника банка — Антипов (А), Борисов (В), Цветков (С) и Дмитриев (D). Известно:

⁵ Задания заимствованы из книги: Андреева Е.В., Босова Л.П., Фалина И.Н. Математические основы информатики. Учебное пособие. М.: БИНОМ. Лаборатория Знаний, 2005, 328 с.

- 1) если A нарушил правила обмена валюты, то и B нарушил;
- 2) если B нарушил, то и C нарушил или A не нарушил;
- 3) если D не нарушил, то A нарушил, а C не нарушил;
- 4) если D нарушил, то и A нарушил.

Кто из подозреваемых нарушил правила обмена валюты?

3. Задача “Пятеро друзей”.

Пятеро друзей решили записаться в кружок любителей логических задач: Андрей (A), Николай (N), Виктор (V), Григорий (G), Дмитрий (D). Но староста кружка поставил им ряд условий: “Вы должны приходиться к нам так, чтобы:

- 1) если A приходит вместе с D , то N должен присутствовать обязательно;
- 2) если D отсутствует, то N должен быть, а V пусть не приходит;
- 3) A и V не могут одновременно ни присутствовать, ни отсутствовать;
- 4) если придет D , то G пусть не приходит;
- 5) если N отсутствует, то D должен присутствовать, но это в том случае, если не присутствует V ; если же и V присутствует при отсутствии N , то D приходиться не должен, а G должен прийти”.

В каком составе друзья смогут прийти на занятия кружка?

4. Брауну, Джонсу и Смиту предъявлено обвинение в соучастии в ограблении банка. Похитители скрылись на поджидавшем их автомобиле. На следствии Браун показал, что преступники скрылись на синем “Бьюике”; Джонс сказал, что это был черный “Крайслер”, а Смит утверждал, что это был “Форд Мустанг” и ни в коем случае не синий. Стало известно, что, желая запутать следствие, каждый из них указал правильно либо только марку машины, либо только ее цвет. Какого цвета и какой марки был автомобиль?

5. Для полярной экспедиции из восьми претендентов — A , B , C , D , E , F , G и H — надо отобрать шестерых специалистов: биолога, гидролога, синоптика, радиста, механика и врача. Обязанности биолога могут выполнять E и G , гидролога — B и F , синоптика — F и G , радиста — C и D , механика — C и H , врача — A и D . Хотя некоторые претенденты владеют двумя специальностями, в экспедиции каждый сможет выполнять только одну обязанность. Кого и кем следует взять в экспедицию, если F не может ехать без B , D — без C и без H , C не может ехать одновременно с G , а A вместе с B ?

Билет № 4

1. Понятие алгоритма: свойства алгоритмов, исполнители алгоритмов. Автоматическое исполнение алгоритма. Способы описания алгоритмов. Основные алгоритмические структуры и их реализация на языке программирования. Оценка эффективности алгоритмов.

Алгоритм — это понятное и точное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи или достижение указанной цели.

Термин имеет интересное историческое происхождение. В IX веке великий узбекский математик *аль-Хорезми* разработал правила арифметических действий над десятичными числами. Совокупность этих правил в Европе стали называть “алгоритм”. Впоследствии слово трансформировалось до известного нам сейчас вида и, кроме того, расширило свое значение: алгоритмом стали называть любую последовательность действий (не только арифметических), которая приводит к решению той или иной задачи. Можно сказать, что понятие вышло за рамки математики и стало применяться в самых различных областях.

Можно выделить три крупных разновидности алгоритмов: **вычислительные, информационные и управляющие**. Первые, как правило, работают с простыми видами данных (числа, вектора, матрицы), но зато процесс вычисления может быть длинным и сложным. Информационные алгоритмы, напротив, реализуют сравнительно небольшие процедуры обработки (например, поиск элементов, удовлетворяющих определенному признаку), но для больших объемов информации. Наконец, управляющие алгоритмы непрерывно анализируют информацию, поступающую от тех или иных источников, и выдают результирующие сигналы, управляющие работой тех или иных устройств. Для этого вида алгоритмов очень существенную роль играет их быстродействие, т.к. управляющие сигналы всегда должны появляться в нужный момент времени.

Каждый алгоритм — это правила, описывающие процесс преобразования исходных данных в необходимый результат. Заметим, что данное важное свойство в некоторых книгах приводят как определение алгоритма.

Для того чтобы произвольное описание последовательности действий было алгоритмом, оно должно обладать следующими свойствами.

• Дискретность

Процесс решения задачи должен быть разбит на последовательность отдельных шагов, каждый из которых называется **командой**. Примером команд могут служить пункты инструкции, нажатие на одну из кнопок пульта управления, рисование графического примитива (линии, дуги и т.п.), оператор языка программирования. Наиболее существенным здесь является тот факт, что алгоритм есть последовательность четко выделенных пунктов — такие “прерывные” объекты в науке принято называть **дискретными**.

• Понятность

Каждая команда алгоритма должна быть понятна тому, кто исполняет алгоритм; в противном случае эта команда и, следовательно, весь алгоритм в целом не могут быть выполнены. Данное требование можно сформулировать более просто и конкретно. Составим полный список команд, который умеет делать исполнитель

алгоритма, и назовем его **системой команд исполнителя** (СКИ).

Требование использовать при составлении алгоритмов только те команды, которые входят в СКИ, связано с тем, что исполнение алгоритма осуществляется формально, без возможности вникнуть в суть команд и проанализировать их.

Одним из таких (вернее, основным из них) “бездушных” исполнителей является ЭВМ. Вообще ЭВМ является универсальным исполнителем алгоритмов. Это связано с тем, что любой алгоритм, составленный для ЭВМ, в конечном итоге транслируется в ее СКИ и, таким образом, становится доступным для исполнения.

• **Определенность (детерминированность)**

Команды, образующие алгоритм (или, можно сказать, входящие в СКИ), должны быть предельно четкими и однозначными. Их результат не может зависеть от какой-либо дополнительной информации извне алгоритма. Сколько бы раз вы не запускали программу, для одних и тех же исходных данных всегда будет получаться один и тот же результат.

При наличии ошибок в алгоритме последнее сформулированное свойство может иногда нарушаться. Например, если не было предусмотрено присвоение переменной начального значения, то результат в некоторых случаях может зависеть от случайного состояния той или иной ячейки памяти компьютера. Но это, скорее, не опровергает, а *подтверждает* правило: алгоритм *должен быть* определенным, в противном случае это не алгоритм.

Определенность также предполагает, что данные, необходимые для выполнения очередной команды алгоритма, получены на одном из предыдущих шагов алгоритма.

• **Результативность (конечность)**

Результат выполнения алгоритма должен быть обязательно получен, т.е. правильный алгоритм не может обрываться безрезультатно из-за какого-либо непреодолимого препятствия в ходе выполнения. Кроме того, любой алгоритм должен завершиться за конечное число шагов. Большинство алгоритмов данным требованиям удовлетворяют, но при наличии ошибок возможны нарушения результативности.

• **Корректность**

Любой алгоритм создан для решения той или иной задачи, поэтому нам необходима уверенность, что это решение будет правильным для любых допустимых исходных данных. Указанное свойство алгоритма принято называть его корректностью. В связи с обсуждаемым свойством большое значение имеет тщательное тестирование алгоритма перед его использованием. Как показывает опыт, грамотная и всесторонняя отладка для сложных алгоритмов часто требует значительно больших усилий, чем собственно разработка этих алгоритмов. При этом важно не столько количество про-

веренных сочетаний входных данных, сколько *количество их типов*. Например, можно сделать сколько угодно проверок для положительных значений аргумента алгоритма, но это никак не будет гарантировать корректную его работу в случае отрицательной величины аргумента.

• **Массовость**

Алгоритм имеет смысл разрабатывать только в том случае, когда он будет применяться многократно для различных наборов исходных данных. Например, если составляется алгоритм обработки текстов, то вряд ли целесообразно ограничивать его возможности только русскими буквами — стоит предусмотреть также латинский алфавит, цифры, знаки препинания и т.п. Тем более что такое обобщение особых трудностей не вызывает.

Таковы основные свойства алгоритмов. Если их внимательно проанализировать, то становится очевидным, что исполнитель алгоритма не нуждается в какой-либо фантазии и сообразительности. Более того, для выполнения алгоритма совсем *не требуется его понимание*, а правильный результат может быть получен путем **формального** и чисто механического следования содержания алгоритма.

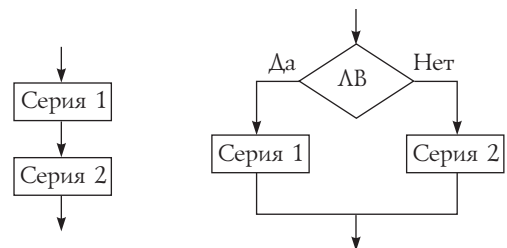
Из возможности формального исполнения алгоритма следует очень важное следствие: поскольку осознавать содержание алгоритма не требуется, его исполнение вполне можно доверить автомату или ЭВМ. Таким образом, **составление алгоритма является обязательным этапом автоматизации любого процесса**. Как только разработан алгоритм, машина может исполнять его лучше человека — быстрее и, что очень важно, не ошибаясь.

Основными способами записи алгоритмов являются:

- словесный;
- словесно-формульный;
- на алгоритмическом языке;
- графический (блок-схема);
- на языке программирования высокого уровня.

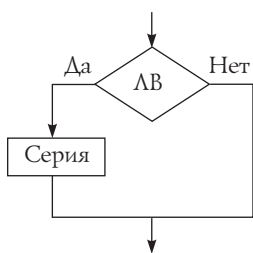
Основными алгоритмическими структурами (ОАС) являются следование, развилка и цикл. В более сложных случаях используются суперпозиции (вложения) ОАС.

Ниже приведены графические обозначения (обозначения на блок-схемах) ОАС.

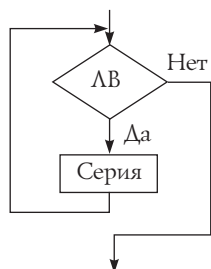
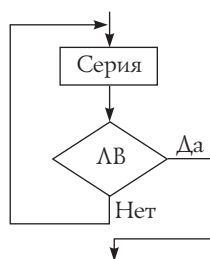
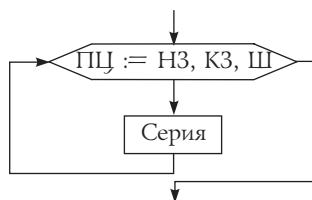


Структура “следование”

Полная развилка



Неполная развилка

Цикл с предусловием
(цикл ПОКА)Цикл с постусловием
(цикл ДО)

Цикл с параметром

На схемах СЕРИЯ обозначает один или несколько любых операторов; ЛВ — логическое выражение (если его значение ИСТИНА, переход происходит по ветви ДА, иначе — по НЕТ). На схеме цикла с параметром использованы обозначения: ПЦ — параметр цикла, НЗ — начальное значение параметра цикла, КЗ — конечное значение параметра цикла, Ш — шаг изменения параметра цикла.

Простейшие задачи имеют *линейный алгоритм* решения. Это означает, что такой алгоритм не содержит проверок условий и повторений, действия в нем выполняются последовательно, одно за другим, т.е. при его реализации используется структура «*следование*».

Чаще всего алгоритмы предполагают обработку некоторых величин. *Величина* — это элемент данных с точки зрения их смыслового (семантического) содержания или обработки. При разработке алгоритма данные можно разбить по смыслу на *входные* — *аргументы*, *выходные* — *результаты*, и *промежуточные*. *Исходные (входные)* — это данные, известные перед выполнением задачи, из условия. *Выходные данные* — результат решения задачи. Переменные, которые не являются ни аргументом, ни результатом алгоритма, а используются только для обозначения вычисляемого промежуточного значения, называются *промежуточными*. Чаще всего требуется указать имена и типы данных — целый, вещественный, логический и символьный, либо структурированный, базирующийся на одном из названных.

Ветвления играют в алгоритмах очень большую роль, поскольку предусматривают корректную реакцию на

самые разнообразные ситуации, возникающие в процессе обработки информации. Благодаря этой структуре алгоритм приобретает способность выбирать один из существующих вариантов работы, наиболее подходящий к сложившейся в данный момент ситуации. В частном случае речь может идти о выполнении или игнорировании при определенных условиях того или иного участка алгоритма.

Значение ветвления в современном программном обеспечении трудно переоценить. Достаточно вспомнить стандартные элементы управления, такие, как меню, радиокнопки, флажки проверки или списки. Именно они дают возможность пользователю чувствовать себя за компьютером свободно и комфортно и выбирать те режимы работы, которые ему нужны.

Приведем также полную форму ветвления в различных алгоритмических языках.

QBasic

IF <ЛВ> **THEN** операторы **ELSE** операторы **ENDIF**

Pascal

IF <ЛВ> **THEN** оператор **ELSE** оператор

C

if (<ЛВ>) оператор; **else** оператор;

Очевидно, что запись отличается лишь незначительными второстепенными деталями. Для получения неполного ветвления ветвь «иначе» разрешается опускать.

Достаточно часто при организации алгоритма решения задачи необходимо одну и ту же определенную последовательность команд выполнить несколько раз подряд. Конечно, самый простой способ — записать эти команды несколько раз друг за другом, и необходимое повторение действий будет организовано. Но как быть в тех случаях, когда количество команд, которые исполняются несколько раз, слишком велико? Или само количество повторений команд огромно? Или вообще неизвестно, а сколько же раз нужно повторить последовательность команд? Решить все эти проблемы можно, если использовать алгоритмическую структуру «цикл».

Командой повторения, или *циклом*, называется такая форма организации действий в алгоритме, при которой выполнение одной и той же последовательности команд повторяется до тех пор, пока истинно некоторое логическое выражение.

Для организации цикла необходимо выполнять следующие действия:

- перед началом цикла задать начальное значение параметров (переменных, используемых в логическом выражении, отвечающем за продолжение или завершение цикла);
- внутри цикла изменять переменную (или переменные), которая сменит значение логического выражения, за счет которого продолжается цикл, на противоположное (для того чтобы цикл в определенный момент завершился);
- вычислять логическое выражение — проверять условие продолжения или окончания цикла;

- выполнять операторы внутри цикла;
- управлять циклом, т.е. переходить к его началу, если он не закончен, или выходить из цикла в противном случае.

Различают циклы с известным числом повторений (цикл с параметром) и итерационные (с пред- и пост-условием).

Опишем схематично, как выполняется каждый из циклов.

Цикл с предусловием:

- вычисляется значение логического выражения;
- если значение логического выражения “истина”, переход к следующему пункту, иначе к п. д);
- выполняется тело цикла;
- переход к п. а);
- конец цикла.

Цикл с постусловием:

- выполняется тело цикла;
- вычисляется значение логического выражения;
- если значение логического выражения “ложь”, переход к п. а), иначе к следующему пункту;
- конец цикла.

Замечание. Таким образом, цикл с постусловием организован, в частности, в алгоритмических языках Pascal и QBasic. В языке C переход к повторению вычислений, как и в цикле с предусловием, осуществляется в случае истинности логического выражения.

Цикл с параметром:

- вычисляются значения выражений, определяющие начальное и конечное значения параметра цикла;
- параметру цикла присваивается начальное значение;
- параметр цикла сравнивается с конечным значением;
- если параметр цикла превосходит (при положительном шаге) конечное значение параметра цикла (или, наоборот, меньше конечного значения параметра цикла при отрицательном шаге), переход к п. з), иначе к следующему пункту;
- выполняется тело цикла;
- параметр цикла автоматически увеличивается на значение шага;
- переход к п. в);
- конец цикла.

Запись циклов на языках программирования

	QBasic	Pascal	C
С предусловием	WHILE AB операторы WEND	WHILE AB DO оператор	while (AB) оператор;
С постусловием	LOOP операторы UNTIL AB	REPEAT операторы UNTIL AB;	do оператор while (AB);
С параметром	FOR ПЦ = НЗ TO КЗ STEP Ш операторы NEXT ПЦ	FOR ПЦ := НЗ TO (DOWNT0) КЗ DO оператор;	for (ПЦ = НЗ; НЗ <= КЗ; ПЦ = ПЦ + Ш) оператор;

Циклы с предусловием и постусловием в большинстве случаев (за исключением отдельных реализаций алгоритмических языков) являются более универсальными по сравнению с циклом с параметром, поскольку в последнем требуется заранее указать число повторений, в то время как в первых двух это не требуется. Цикл с параметром в любом случае может быть преобразован к циклу с пред- или постусловием. Обратное верно не всегда.

Замечание. В языке C цикл for на самом деле является универсальным циклом с предусловием. В частности, из него можно сделать и описанную форму цикла с параметром.

Примеры использования основных алгоритмических структур и их суперпозиций для составления алгоритмов, а по ним — программ приводятся в многочисленной литературе по программированию, поэтому не будем на них здесь останавливаться.

Наконец, рассмотрим вопрос анализа алгоритмов.

Одну и ту же задачу могут решать много алгоритмов. Эффективность работы каждого из них описывается разнообразными характеристиками. Прежде чем анализировать эффективность алгоритма, нужно доказать, что данный алгоритм правильно решает задачу. В противном случае вопрос об эффективности не имеет смысла. Если алгоритм решает поставленную задачу, то можно посмотреть, насколько это решение эффективно.

При анализе алгоритма определяется количество “времени”, необходимое для его выполнения. Это не реальное число секунд или других промежутков времени, а приблизительное число операций, выполняемых алгоритмом. Число операций и измеряет относительное время выполнения алгоритма. Таким образом, иногда “временем” называют вычислительную сложность алгоритма. Фактическое количество секунд, требуемое для выполнения алгоритма на компьютере, непригодно для анализа, т.к. обычно интересует только относительная эффективность алгоритма, решающего конкретную задачу. Действительно, время, требуемое на решение задачи, — не очень хороший способ измерять эффективность алгоритма, потому что алгоритм не становится лучше, если его перенести на более быстрый компьютер, или хуже, если его исполнять на более медленном.

На самом деле фактическое количество операций алгоритма на тех или иных входных данных не представляет большого интереса и не очень много сообщает об алгоритме. Реально определяется зависимость числа операций конкретного алгоритма от размера входных данных. Можно сравнить два алгоритма по скорости роста числа операций. Именно скорость роста играет ключевую роль, поскольку при небольшом размере входных данных алгоритм А может требовать меньшего количества операций, чем алгоритм В, но при росте объема входных данных ситуация может поменяться на противоположную.

Два самых больших класса алгоритмов — это алгоритмы с повторением и рекурсивные алгоритмы. В основе алгоритмов с повторением лежат циклы и условные выражения; для

анализа таких алгоритмов требуется оценить число операций, выполняемых внутри цикла, и число итераций цикла. Рекурсивные алгоритмы разбивают большую задачу на фрагменты и применяются к каждому фрагменту по отдельности. Такие алгоритмы называются иногда “разделяй и властвуй”, и их использование может оказаться очень эффективным. В процессе решения большой задачи путем деления ее на меньшие создаются небольшие, простые и понятные алгоритмы. Анализ рекурсивного алгоритма требует подсчета количества операций, необходимых для разбиения задачи на части, выполнения алгоритма на каждой из частей и объединения отдельных результатов для решения задачи в целом. Объединяя эту информацию и информацию о числе частей и их размере, можно вывести рекуррентное соотношение для сложности алгоритма. Полученному рекуррентному соотношению можно придать замкнутый вид, затем сравнивать результат с другими выражениями.

Приведем пример оценки сложности алгоритма.

Рассмотрим известный алгоритм сортировки выбором (здесь он записан в виде функции на C++) и оценим его сложность по описанной выше методике.

```
void v(int n, int a[1000]) {
    int i, j, vs, m;
    for (i = 0; i < n - 1; i++) {
        m = i;
        for (j = i + 1; j < n; j++)
            if (a[j] < a[m]) m = j;
        vs = a[i];
        a[i] = a[m];
        a[m] = vs;
    }
}
```

Для каждой отдельной операции введем понятие “стоимости” (т.е. времени ее выполнения), в конечном итоге оценим, как зависит “время” выполнения алгоритма от размера сортируемого массива n .

Суммируем

$$S = c_1(n-1) + c_2(n-1) + c_3(n-1) + c_1(n-1) + c_1n^2 + c_2n^2 + c_3n^2 + c_1n^2 + c_2n^2 + c_1(n-1) + c_1(n-1) + c_1(n-1) = (2c_1 + 2c_2 + c_3)n^2 + (4c_1 + c_2 + c_3)n - (c_1 + c_2 + c_3)$$

Таким образом, замечаем, что наибольший вклад в количество операций при больших n вносит величина n^2 , т.е. сложность алгоритма пропорциональна n^2 .

Использованные источники информации

1. *Семакин И., Залогова А., Русаков С., Шестакова А.* Информатика: учебник по базовому курсу. М.: Лаборатория Базовых Знаний, 1998. (Глава 12. Введение в программирование, с. 323–371.)

2. *Угринович Н.* Информатика и информационные технологии. Учебное пособие для общеобразовательных учреждений. М.: БИНОМ, 2001, 464 с.

3. Информатика. 7–8-е классы / Под ред. Н.В. Макаровой. СПб.: ПитерКом, 1999, 368 с.

Операторы	Стоимость	Сложность
void v(int n, int a[1000])		
{int i, j, vs, m;		
for (i = 0; i < n - 1; i++)	c_1, c_2, c_3	$c_1(n-1) + c_2(n-1) + c_3(n-1)$
{m = i;	c_1	$c_1(n-1)$
for (j = i + 1; j < n; j++)	c_1, c_2, c_3	$c_1n^2 + c_2n^2 + c_3n^2$
if (a[j] < a[m]) m = j;	c_2, c_3	$c_2n^2 + c_3n^2$
vs = a[i];	c_1	$c_1(n-1)$
a[i] = a[m];	c_1	$c_1(n-1)$
a[m] = vs;}	c_1	$c_1(n-1)$

4. *Шафрин Ю.А.* Информационные технологии. М.: Лаборатория Базовых Знаний, 1998, 704 с. (п. 1.6. Понятие об алгоритмах, п. 1.7. Понятие о программировании, с. 53–72).

5. Информатика. Задачник-практикум в 2 т. / Под ред. И.Г. Семакина, Е.К. Хеннера: Т. 1. М.: Лаборатория Базовых Знаний, 1999, 304 с.

6. Основы информатики и вычислительной техники. Пробное учебное пособие для средних учебных заведений / Под ред. А.П. Ершова, В.М. Монахова. М.: Просвещение, 1985. Ч. I, II.

7. *Шауцукова А.З.* Информатика: Учебник для 10–11-х классов. М.: Просвещение, 2000 г. (Глава 7. Алгоритмы. Алгоритмизация. Алгоритмические языки.)

7. http://cjmp-science.narod.ru/didakt_i.html — дидактические и методические материалы по программированию и информатике.

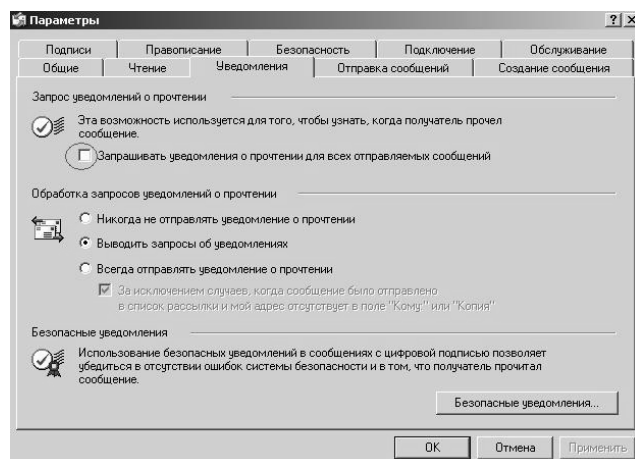
8. *Макконнелл Дж.* Анализ алгоритмов. Вводный курс. М.: Техносфера, 2002, 304 с.

2. Средствами почтовой программы обеспечить автоматическое уведомление отправителя о получении от него письма

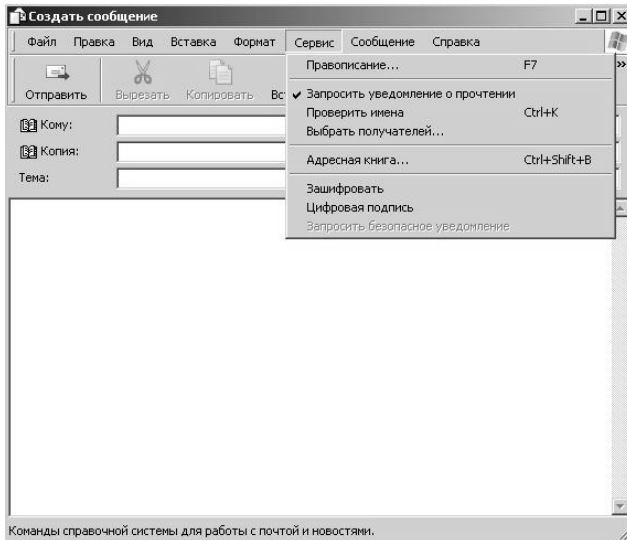
Рассмотрим решение поставленной задачи в двух разных почтовых клиентах.

Outlook Express

В данной программе можно настроить получение уведомления о прочтении письма для каждого отправленного сообщения (настройка с помощью меню Сервис > Параметры > Уведомления).

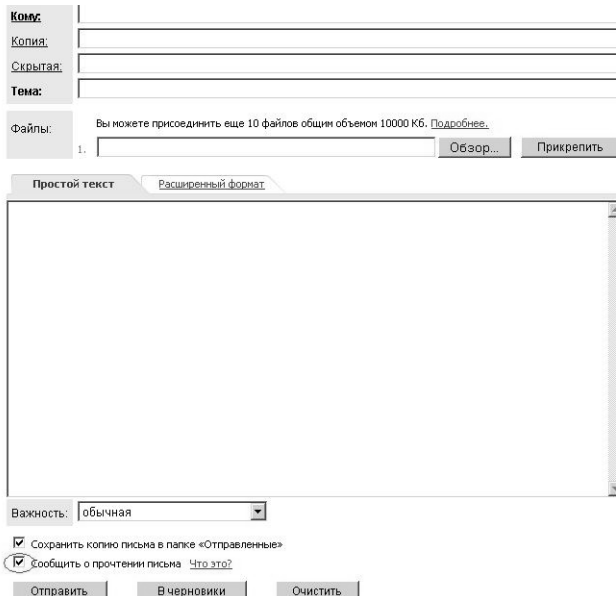


Если уведомление требуется не часто, то эту настройку можно не осуществлять, а запросить уведомление непосредственно при создании сообщения.



Mail.ru

При создании сообщения можно установить извещение о получении этого сообщения.



В других почтовых клиентах и бесплатных почтовых сервисах существует аналогичная возможность.

3. Подсчитать информационный объем графического файла по размеру в пикселях с учетом палитры (заданы количество цветов в палитре и размер рисунка) и того же рисунка в графическом формате со сжатием (задан коэффициент сжатия).

Задан рисунок размером 1024×768 с представлением информации в формате RGB. Определить информаци-

онный объем графического файла, хранящего такой рисунок. Вычислить объем файла в том случае, если для хранения этого же рисунка используется формат со сжатием с коэффициентом 0,2.

Решение

В формате RGB для представления каждого пикселя используется три байта (по одному на каждый базовый цвет).

Таким образом, получаем

$$1024 \times 768 \times 3 \text{ байта} = 768 \times 3 \text{ Кб} = 2,25 \text{ Мб}$$

При использовании формата со сжатием получаем $2,25 \text{ Мб} \times 0,8 = 1,8 \text{ Мб}$

Очередная брошюра — подписчикам бесплатно

В октябре все подписчики "Информатики" бесплатно получают очередную брошюру Библиотечки "Первого сентября", серии "Информатика" "Эффективность алгоритмов".

Автор брошюры, Яков Наумович Зайдельман, хорошо знаком не только нашим читателям, но и всей стране по выступлениям в популярной интеллектуальной телепередаче "Своя игра".

В брошюре на наглядных и несложных задачах обсуждаются интересные и нетривиальные вопросы, связанные с эффективностью алгоритмов.

Как сравнить эффективность алгоритмов? Всегда ли существует эффективный алгоритм? Стоит ли вообще задумываться над вопросами эффективности, когда мощности современных компьютеров позволяют решать "в лоб" даже весьма громоздкие задачи?

Надеемся, брошюра не только доставит вам удовольствие, но и не раз пригодится при подготовке к урокам.



Уважаемые коллеги!

Примерные ответы на профильные билеты будут публиковаться по два в номере. Мы не успеем завершить публикации в этом полугодии, и они продолжатся в январе. Пожалуйста, учитывайте это и не забудьте оформить подписку на первое полугодие 2007 г.

Азы информатики.

Выходим в Интернет

А.А. ДУВАНОВ,

г. Переславль-Залесский,
kurs@robotland.pereslavl.ru

Продолжение. Начало см. в № 17–19/2006

Вопросы и ответы Академии-2

Объясните смысл терминов

1. Ethernet, Fast Ethernet, Gigabit Ethernet

Ответ. Ethernet, Fast Ethernet, Gigabit Ethernet — технологии локальных сетей с разделяемой средой. Различаются скоростями передачи данных (10 Мбит/с, 100 Мбит/с, 1000 Мбит/с), максимальным диаметром (2500 м, 200 м, 200 м), другими параметрами, но принцип работы один и тот же:

- узлы сети связаны единой передающей средой;
- узел может начать свою передачу, если по сети не передаются данные другими узлами;
- после передачи пакета (длина от 64 до 1518 байт) узел выдерживает паузу перед передачей следующего (чтобы дать возможность работать другим узлам);
- передача пакета прекращается, если обнаружена коллизия;
- передача пакета повторяется через случайное время из фиксированного диапазона (от 0 до 52,4 мс).

В стандарте Gigabit Ethernet увеличена длина минимального пакета с 64 до 512 байт и, кроме того, разрешено передавать несколько пакетов подряд без пауз (общий размер — не более 8192 байт).

2. 10G Ethernet

Ответ. Технология построения сети 10G Ethernet (10 000 Мбит/с) принципиально отличается от других Ethernet-технологий. Рабочие станции в такой сети не разделяют общую среду. Сети 10G Ethernet — это сети с коммутацией пакетов.

3. Разделяемая среда

Ответ. Разделяемая среда — это такой способ организации работы сети, при котором сообщение от одной рабочей станции достигает всех других при помощи одного общего канала связи. В качестве каналов связи традиционно используются кабели: коаксиальный, витая пара, оптоволоконный. Но может быть использована инфракрасная или радиосвязь (в том числе через спутник).

Примером общения посредством разделяемой среды может служить обычный разговор в компании вежливых людей (метафора С.А. Абрамова). Человек начинает говорить лишь тогда, когда все другие молчат. Реплику каждого человека слышат все, никто не перебивает говорящего и не вникает в смысл чужого сообщения. Никто не говорит слишком долго (не монополизирует право на

разговор). Если по недоразумению говорить начинают несколько собеседников (коллизия), все умолкают, повторяя попытку через некоторое (случайное) время.

4. Терминатор

Ответ. Терминатор — это электронные заглушки на концах сегмента кабеля в разделяемых средах. Назначение терминатора — поглощение сигнала, распространяемого по кабелю.

5. Кадр

Ответ. Кадрами в терминологии Ethernet называют пакеты, на которые разделяют передаваемое сообщение. Дробить сообщение на порции необходимо, чтобы предотвратить монополизацию разделяемой среды одной рабочей станцией.

Кадр, кроме фрагмента информации, содержит MAC-адреса отправителя и получателя, контрольную сумму (для проверки сохранности пакета в пункте получения) и другую служебную информацию.

В стандартах Ethernet и Fast Ethernet длина кадра может принимать значение от 64 до 1518 байт. В стандарте Gigabit Ethernet минимальный размер кадра увеличен до 512 байт и, кроме того, разрешается передавать несколько кадров подряд без пауз (общий размер — не более 8192 байт).

6. MAC-адрес

Ответ. MAC-адрес (*Media Access Control* — управление доступом к носителю) — 48-битное число, которое однозначно идентифицирует сетевое электронное устройство.

Уникальный MAC-адрес “зашивается” в сетевой адаптер при его изготовлении. Он не может совпадать ни с каким другим MAC-адресом в мире и не может меняться во время эксплуатации устройства.

Распределением MAC-адресов между производителями оборудования занимается международная некоммерческая организация IEEE (*Institute of Electrical and Electronics Engineers* — Институт инженеров электротехники и электроники).

7. Коллизия

Ответ. Коллизия — наложение сигналов от двух (или более) передающих станций в разделяемой среде передачи.

8. Ранняя коллизия

Ответ. Ранней называется такая коллизия, которую передающая станция способна распознать до окончания передачи пакета.

9. Поздняя коллизия

Ответ. Коллизия называется поздней, если она возникает после завершения передачи пакета, вызвавшего коллизиию.

10. Диаметр сети

Ответ. Диаметр сети — максимальное расстояние между двумя узлами в сети, вычисляемое по совокупной длине соединяющих станции каналов связи.

Расстояние между компьютерами не всегда совпадает с длиной каналов связи. Два компьютера могут находиться по разные стороны одной стены. Физическое расстояние между ними — менее одного метра, а длина соединительного кабеля может измеряться десятками метров, если по каким-то причинам (например, музейного свойства) стену нельзя сверлить.

11. Повторитель

Ответ. Повторитель (repeater) — это простое электронное устройство (без всякого программного обеспечения), которое усиливает сигнал при передаче его из одного сегмента кабеля в другой.

На рис. 3.21 показана сеть, в которой кабель состоит из трех сегментов, соединенных двумя повторителями:

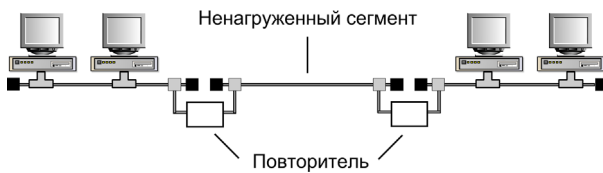


Рис. 3.21. Сеть с повторителями

12. Толстый коаксиал

Ответ. Коаксиальный кабель диаметром 0,5 дюйма (рис. 3.22).



Рис. 3.22. Коаксиальный кабель

Коаксиальный кабель устроен так же, как телевизионный кабель: в центре — медная жила (или несколько жил), затем изоляция, затем металлическая оплетка, наконец — внешний слой изоляции.

13. Тонкий коаксиал

Ответ. Коаксиальный кабель диаметром 0,25 дюйма.

14. Витая пара

Ответ. Витая пара — два изолированных (медных) проводника, скрученные один относительно другого с постоянным шагом (рис. 3.23).

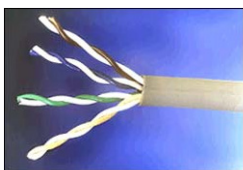


Рис. 3.23. Кабель витая пара

Обычно под витой парой понимают кабель, в общей изоляционной оболочке которого расположено несколько витых пар (2 или 4).

15. Оптоволоконно

Ответ. Основа оптоволоконного кабеля — нить из стекла или пластмассы, по которой передается световой луч (рис. 3.24). Оптоволоконная жила окружена непрозрачной зеркальной оболочкой. Благодаря этому свет, распространяющийся внутри волокна, испытывает полное внутреннее отражение от оболочки и не покидает волокна даже тогда, когда оно делает изгибы.



Рис. 3.24. Оптоволоконный кабель

16. 10Base-5, 10Base-2, 10Base-T, 10Base-F

Ответ. Это варианты стандарта Ethernet, основанные на свойствах физической среды передачи данных.

- 10Base-5 — коаксиальный кабель диаметром 0,5 дюйма, называемый “толстым”.
- 10Base-2 — коаксиальный кабель диаметром 0,25 дюйма, называемый “тонким”.
- 10Base-T — неэкранированная витая пара (две пары в кабеле).
- 10Base-F — волоконно-оптический кабель.

Число 10 в указанных обозначениях обозначает битовую скорость передачи в этих стандартах — 10 Мбит/с.

Ниже приводится сравнительная таблица этих стандартов.

	10Base-5	10Base-2	10Base-T	10Base-F
Среда передачи	Толстый коаксиал	Тонкий коаксиал	Витая пара	Оптоволоконно
Максимальная длина сегмента	500 м	185 м	100 м	2000 м
Максимальный диаметр сети	2500 м	925 м	500 м	2500 м
Максимальное число станций	1024	1024	1024	1024

17. 100Base-TX, 100Base-T4, 100Base-FX

Ответ. Это варианты стандарта Fast Ethernet, основанные на свойствах физической среды передачи данных (коаксиальный кабель исключен из списка разрешенных сред передачи):

- 100Base-TX — неэкранированная или экранированная витая пара (две пары в кабеле).

- 100Base-T4 — неэкранированная витая пара (четыре пары в кабеле).
- 100Base-FX — волоконно-оптический кабель (с двумя волокнами).

Число 100 в указанных обозначениях обозначает битовую скорость передачи в этих стандартах — 100 Мбит/с.

18. Трансивер

Ответ. Трансивер — это электронное устройство, которое методом прокалывания соединяют с толстым коаксиалом, а при помощи витой пары (длиной до 50 м) — с сетевым адаптером. Коаксиальный кабель диаметра 0,5 дюйма такой “толстый”, что его сложно подсоединять непосредственно к сетевому адаптеру компьютера (рис. 3.25).

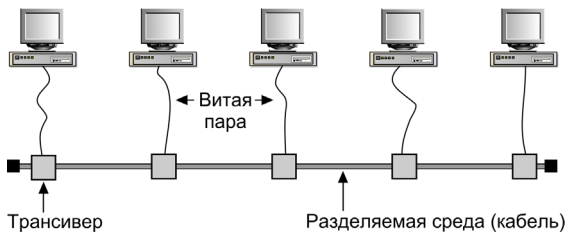


Рис. 3.25. Сеть на толстом коаксиале с трансиверами

Трансивер — это не просто механический соединитель (как Т-образный разъем для тонкого коаксиала). Фактически трансивер — это часть сетевого адаптера, вынесенного прямо на кабель (рис. 3.26).

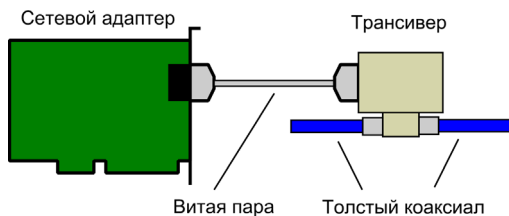


Рис. 3.26. Трансивер — часть сетевого адаптера, вынесенного на кабель

19. Т-образный разъем

Ответ. Этот разъем используется для подсоединения тонкого коаксиала к сетевой карте компьютера (рис. 3.27).

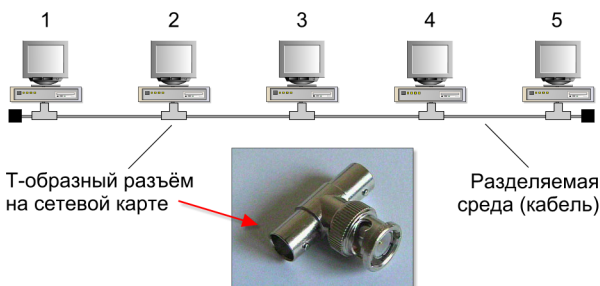


Рис. 3.27. Сеть на тонком коаксиале с Т-образными разъемами

20. Хаб (многопортовый повторитель, концентратор)

Ответ. Хаб (hub, буквально — ступица колеса) — сетевое устройство, объединяющее несколько узлов сети Ethernet в общий фрагмент одной разделяемой среды.

Сетевой кабель подсоединяется к хабу (рис. 3.28) при помощи *портов* (соединительных разъемов):



Рис. 3.28. Хаб

Устройства подключаются к хабу при помощи витой пары, коаксиального кабеля или оптоволокну. Порты хаба должны быть оборудованы соответствующими разъемами.

На рис. 3.29 показана сеть с хабом, у которого четыре порта. К каждому порту подсоединен сетевой адаптер рабочей станции.

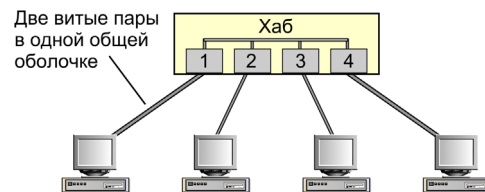


Рис. 3.29. Сеть звезда с хабом в центре

Хаб не содержит никакого программного обеспечения (в том числе он не занимается маршрутизацией пакетов), он просто соединяет порты и усиливает сигнал, передавая его с одного порта на все остальные.

21. Правило 5–4–3

Ответ. Правило 5–4–3 — это правило построения сетей Ethernet на коаксиальном кабеле: пять сегментов, четыре повторителя, три нагруженных сегмента.

То есть разрешается использовать в сети не более 4 повторителей и, соответственно, не более 5 сегментов кабеля. Причем только 3 сегмента из 5 могут быть нагруженными (с подключенными рабочими станциями). Между нагруженными сегментами должны быть ненагруженные. Максимальная конфигурация сети показана на рис. 3.30.

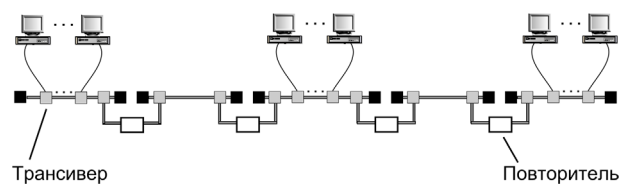
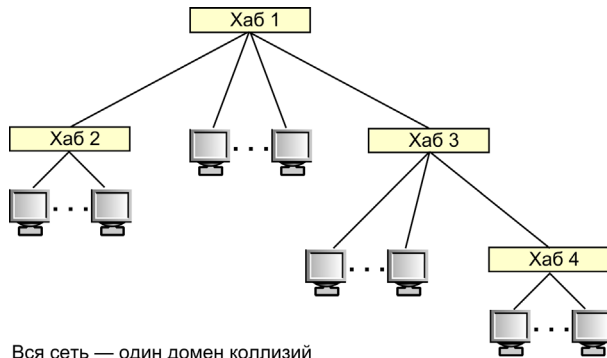


Рис. 3.30. Максимальная конфигурация сети с повторителями

22. Правило 4 хабов

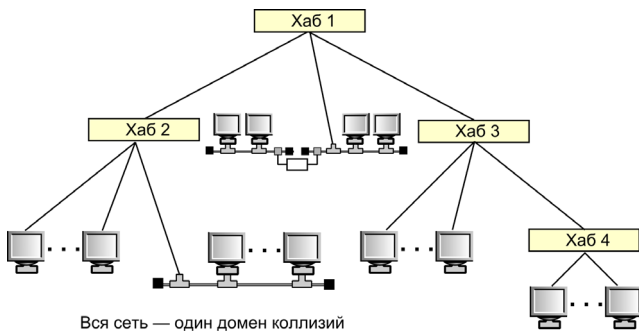
Ответ. Правило 4 хабов — это правило построения сети (или подсети) с одним доменом коллизий: максимальное количество хабов между любыми двумя станциями не должно быть больше четырех. Сеть на рис. 3.31 удовлетворяет этому требованию.



Вся сеть — один домен коллизий

Рис. 3.31. Сеть, удовлетворяющая правилу 4 хабов

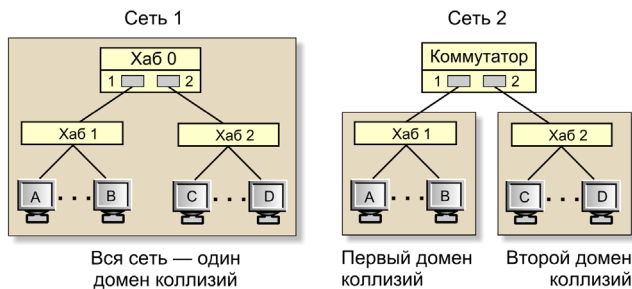
Если при построении сети используются и повторители, и хабы, то при проверке правила 4 хабов повторитель приравнивается к хабу (фактически повторитель и есть хаб с двумя портами). Сеть на рис. 3.32 тоже удовлетворяет правилу 4 хабов:



Вся сеть — один домен коллизий

Рис. 3.32. Сеть, удовлетворяющая правилу 4 хабов

Когда сеть при помощи коммутаторов или маршрутизаторов разбита на несколько доменов коллизий, правило 4 хабов работает независимо в каждом домене, но не относится ко всей сети в целом (рис. 3.33).



Вся сеть — один домен коллизий

Первый домен коллизий

Второй домен коллизий

Рис. 3.33. В сети 1 — один домен коллизий, в сети 2 — два.

23. Коммутатор (мост, переключатель)

Ответ. Коммутатор (рис. 3.34) — сетевое устройство, которое, подобно хабу, соединяет несколько узлов сети Ethernet, но в отличие от хаба разбивает сеть своими портами на несколько отдельных доменов коллизий.

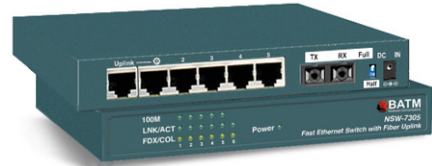


Рис. 3.34. Коммутатор

Происходит так потому, что коммутатор в отличие от хаба не транслирует полученный пакет на другие порты, если получатель находится на том же самом порту, с которого получен пакет.

Коммутатор, кроме электроники, содержит программное обеспечение, которое управляет его работой, в частности, автоматически строит таблицу маршрутизации.

24. Домен коллизий

Ответ. Домен коллизий — фрагмент сети с общей разделяемой средой.

Сеть, построенная на общей шине, содержит единый домен коллизий (рис. 3.35).

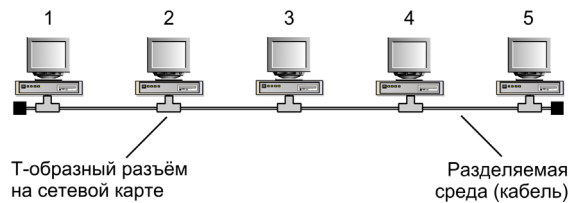


Рис. 3.35. Сеть на общей шине

Повторители, соединяющие фрагменты коаксиального кабеля, не разделяют сеть на домены коллизий — разделяемая среда по-прежнему одна (рис. 3.36).

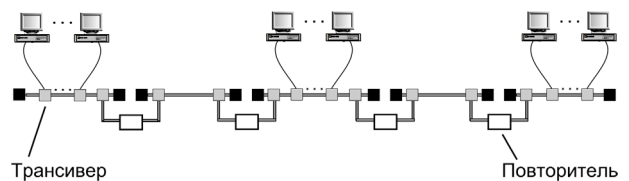
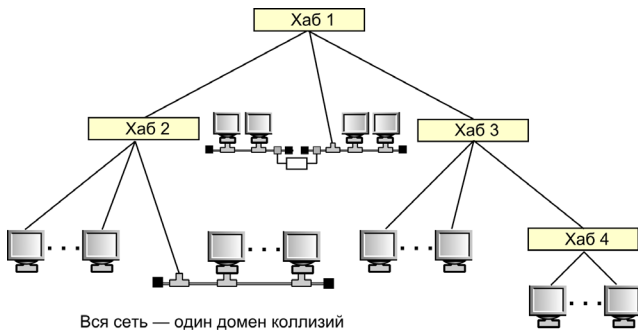


Рис. 3.36. Сеть с повторителями

Сеть, построенная только на повторителях и хабах, представляет собой единый домен коллизий (рис. 3.37).



Вся сеть — один домен коллизий

Рис. 3.37. Сеть с повторителями и хабами

Коммутаторы и маршрутизаторы делят сеть на независимые домены коллизий (рис. 3.38).

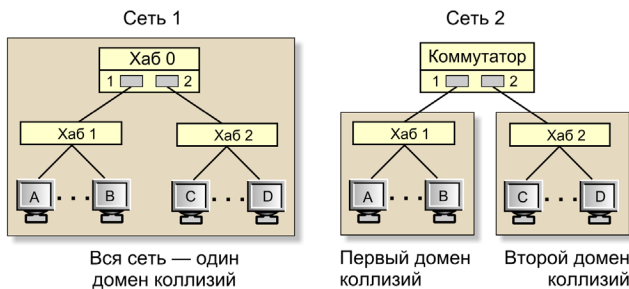


Рис. 3.38. В сети 1 — один домен коллизий, в сети 2 — два.

25. Таблица маршрутизации

Ответ. Таблица маршрутизации — таблица коммутатора (или маршрутизатора), в которой строки определяют порты, на которых находятся источники сетевых пакетов.

Пусть сеть имеет вид, изображенный на рис. 3.39.

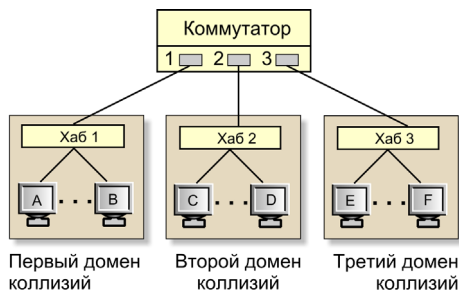


Рис. 3.39. Пример сети с коммутатором

Если считать, что работают только станции А, В, С, D и F, то таблица маршрутизации для единственного коммутатора в этой сети будет иметь вид:

Адрес узла	Порт, по которому находится узел
A	1
B	1
C	2
D	2
E	3
F	3

26. Маршрутизатор (роутер)

Ответ. Маршрутизатор — сетевое устройство, которое, подобно коммутатору, соединяет (коммутирует) узлы сети в том случае, когда это необходимо для передачи пакета.

Маршрутизаторы — более сложные устройства по сравнению с коммутаторами. Они более надежно изолируют трафики отдельных частей сети и, главное, способны работать в сети с ячеистой топологией, обеспечивая выбор наиболее рационального маршрута. В то время как повторители, хабы и коммутаторы способны работать только в сети с единственным маршрутом между любыми двумя узлами (например, в сети с иерархической структурой).

27. Сеть с разделяемой средой

Ответ. Сеть с разделяемой средой — это сеть, в которой каждый узел получает все, что передается по сети; передачу выполняет только один узел, остальные ждут паузы для начала собственной передачи. Если в сети возникает коллизия, узлы начинают повторную передачу испорченного пакета через случайный промежуток времени из фиксированного интервала.

28. Сеть с коммутацией пакетов

Ответ. Сеть с коммутацией пакетов — это сеть, в которой пакеты не “разбрасываются” по всей сети, а целенаправленно “проталкиваются” от узла к узлу по направлению к пункту назначения. За продвижение пакетов в такой сети отвечают маршрутизаторы. Они определяют соседний узел, в который нужно передвинуть пакет для приближения его к пункту назначения.

Ответьте на вопросы

1. Как работает сеть с разделяемой средой?

Ответ. В сетях с разделяемой средой работа выполняется по следующему алгоритму:

1. Если в сети “тишина”, можно начать передачу пакета.
2. Если обнаружена коллизия, нужно прекратить передачу.
3. Через случайную паузу нужно повторить передачу испорченного пакета.

2. Почему в сети с разделяемой средой сообщение для передачи разделяется на пакеты (кадры)?

Ответ. Разделение сообщения на пакеты предотвращает монополизацию общей среды одним узлом. После передачи пакета узел должен сделать паузу, которой может воспользоваться другой узел и начать свою передачу.

3. Как рабочая станция узнает, что передаваемый по сети пакет предназначен для нее?

Ответ. Каждый пакет, кроме фрагмента передаваемого сообщения, содержит MAC-адрес получателя. Рабочие станции сравнивают адрес из пакета со своим собственным адресом и в случае совпадения принимают пакет полностью.

4. Кто и когда назначает MAC-адреса устройствам, входящим в Ethernet-сеть?

Ответ. Уникальный MAC-адрес “зашивается” в сетевой адаптер при его изготовлении. Он не может совпадать ни с каким другим MAC-адресом в мире и не может меняться во время эксплуатации устройства.

Распределением MAC-адресов между производителями оборудования занимается международная некоммерческая организация IEEE (*Institute of Electrical and Electronics Engineers* — Институт инженеров электротехники и электроники).

5. Протокол Ethernet запрещает начинать передачу, если в сети присутствует сигнал. Почему же в разделяемых средах возникают коллизии?

Ответ. Сигнал распространяется по среде с конечной скоростью и может не успеть дойти до станции, которая начинает передачу, решив, что сеть свободна. Возникает коллизия.

6. Является ли коллизия исключительной ситуацией в сети с разделяемой средой?

Ответ. В сети с разделяемой средой коллизия является обычной рабочей ситуацией.

7. За счет какого приема протокол Ethernet обеспечивает работоспособность сети, несмотря на коллизии?

Ответ. При обнаружении коллизии станции должны прекратить передачу. Прием, который обеспечивает работоспособность сети, — случайная пауза для возобновления передачи пакета, испорченного коллизией.

8. Почему ранняя коллизия не приводит к потерям пакетов?

Ответ. Коллизия, которую передающая станция обнаруживает до окончания передачи пакета, называется ранней. Станция своевременно фиксирует неудачную передачу и возобновляет ее через случайную паузу. Испорченный пакет посылается заново.

9. Почему поздняя коллизия приводит к потерям пакетов?

Ответ. Коллизия, которая возникает после окончания передачи пакета, называется поздней. Пакет уже передан в сеть, ответственность за него снята с передающей

станцией, но коллизия искажает пакет и станция назначения получает его испорченным (или не получает вовсе, если в пакете искажен MAC-адрес получателя).

10. Что является главной причиной ограничения диаметра сети в сети с разделяемой средой?

Ответ. Главная причина, по которой ограничивают диаметр сети с разделяемой средой, — предотвращение поздних коллизий. Чем длиннее сеть, тем больше времени нужно сигналу для ее прохождения, тем больше вероятность поздних коллизий (коллизий, возникших после того, как пакет полностью передан в сеть передающей станцией).

11. Как можно вычислить максимальный диаметр сети с разделяемой средой?

Ответ. Для предотвращения поздних коллизий диаметр сети ограничивают величиной, при которой время передачи пакета наименьшей длины было бы больше удвоенного времени прохождения сигнала по всей длине кабеля.

Диаметры сетей для разных стандартов Ethernet указываются с большим запасом (более чем в три раза).

Оценим максимальный диаметр сети Ethernet.

Исходные данные:

Скорость передачи данных:

$$V_d = 10 \text{ Мбит/с} = 10^7 \text{ бит/с}$$

Скорость сигнала: $V_s = 300\,000 \text{ км/с} = 3 \cdot 10^8 \text{ м/с}$

Длина минимального пакета:

$$L_m = 64 \text{ байта} = 512 \text{ бит}$$

Определим время передачи пакета:

$$T = L_m / V_d = 512 / 10^7 \text{ (сек.)}$$

Определим расстояние, которое пройдет сигнал по кабелю за это время:

$$S = V_s \cdot T = 3 \cdot 10^8 \cdot 512 / 10^7 = 3 \cdot 10 \cdot 512 = 15\,360 \text{ (м)}$$

Удвоенная длина кабеля должна быть меньше этого значения, то есть кабель должен быть короче:

$$15\,360 / 2 = 7680 \text{ м.}$$

По стандарту Ethernet максимальный диаметр сети не должен превышать 2500 м. Видим, что стандарт указывает значение более чем с трехкратным запасом.

12. Почему в расчетах максимального диаметра учитывается удвоенное время прохождения сигнала по всей длине сетевого кабеля?

Ответ. Учитывается крайний случай, когда станции, вызвавшие коллизия, расположены на противоположных концах кабеля, и коллизия возникает в момент, когда сигнал от первой станции прошел почти всю длину кабеля (за время t).

Передающая станция обнаружит коллизия, когда искаженный сигнал вернется к ней (через то же время t).

Таким образом, для предотвращения поздних коллизий длина кабеля должна быть такой, чтобы удвоенное время прохождения сигнала по всей его длине ($2t$) было меньше времени передачи пакета минимальной длины.

13. Можно ли увеличить диаметр сети, увеличив минимальную длину пакета?

Ответ. Да, конечно. Удлинение пакета (L_m) увеличивает время его передачи (T), а значит, увеличивает расстояние, которое пройдет сигнал по кабелю за это время:

$$T = L_m / V_d$$

$$S = V_s \cdot T$$

Именно так и поступают в стандарте Gigabit Ethernet. В этом стандарте длина минимального пакета увеличена с 64 до 512 байт, что позволяет увеличить максимальный диаметр сети в 8 раз ($512/64$).

14. Почему для сети с разделяемой средой стандарты предусматривают ограничение на число подключаемых к ней узлов?

Ответ. При большом количестве узлов дождаться паузы в сети для начала передачи может оказаться не просто. Стандарты называют такое количество узлов, при котором сеть остается работоспособной даже при максимальной нагрузке (когда все узлы работают одновременно).

15. При помощи каких устройств можно построить Ethernet-сеть с древовидной топологией?

Ответ. Хабсы, коммутаторы, маршрутизаторы.

16. Какие преимущества имеет древовидная сеть по сравнению с сетью, построенной на общей шине?

Ответ. Древовидная сеть более надежна. Вывод из строя любого узла отражается только на его потомках и не мешает работе остальной части сети (рис. 3.40).

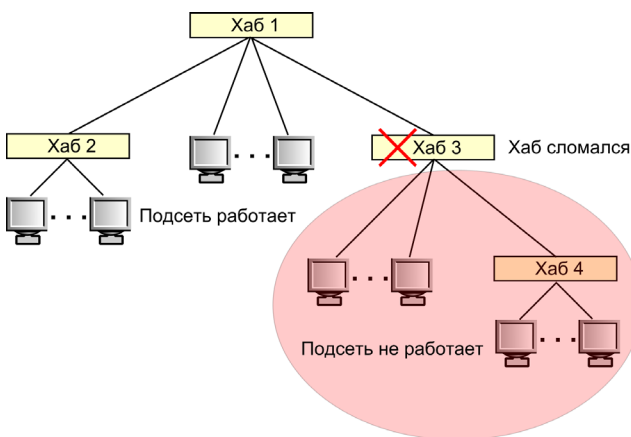


Рис. 3.40. Вывод из строя узла отражается только на его потомках

В сети с общей шиной разрыв кабеля в любом месте приводит к краху сети в целом (рис. 3.41).



Рис. 3.41. Вывод из строя одного узла приводит к краху сети в целом

В древовидной сети на хабсах (коммутаторах или маршрутизаторах) число физических соединений меньше, чем в сети с общей шиной, значит, и в силу этого надежность древовидной сети выше (для подключения к общему кабелю нужно три контактных точки, для подсоединения к хабу — две) (рис. 3.42).

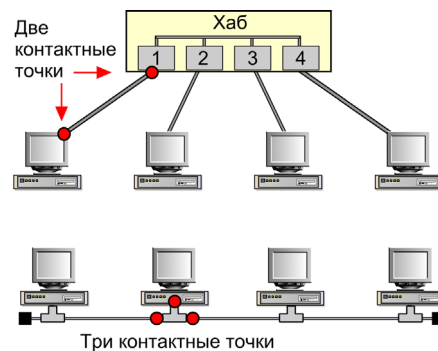


Рис. 3.42. Подключение к общему кабелю и хабу

17. Говорят, что Ethernet-сеть, в которой физические соединения рабочих станций при помощи хабсов образуют древовидную структуру, логически эквивалентна сети с общей шиной. Что это означает?

Ответ. Несмотря на то что физические соединения в сети с хабами образуют дерево, принципиально сеть не отличается от сети с общей шиной: хабы объединяют рабочие станции общей разделяемой средой. Поэтому говорят: топология физических связей в такой сети — дерево, логическое соединение — общая шина (одна разделяемая среда).

18. Эквивалентна ли древовидная Ethernet-сеть, в которой физические соединения рабочих станций выполнены при помощи коммутаторов, сети с общей шиной?

Ответ. Нет. Коммутаторы своими портами разбивают сеть на соответствующее число доменов коллизий: передача внутри одного домена не попадает на другие, а передачу в общей шине “слышат” все станции.

19. Эквивалентна ли древовидная сеть, в которой физические соединения рабочих станций выполнены при помощи маршрутизаторов, сети с общей шиной?

Ответ. Нет. По той же причине, что и в сети с коммутаторами. Маршрутизатор, как и коммутатор, внутренние сообщения не транслирует на всю сеть.

20. Решает ли хаб (коммутатор, маршрутизатор) проблему коллизии?

Ответ. Хаб — нет. Хаб транслирует сигнал, полученный с одного порта на все другие, то есть работа сети с хабами эквивалентна работе сети с общей шиной.

Коммутатор и маршрутизатор — частично решают, разбивая сеть на несколько доменов коллизий. Все ограничения, связанные с поздними коллизиями, применяются теперь не ко всей сети в целом (как в сети с хабами), а отдельно к каждому домену (подключенному к порту устройству).

21. Может ли сеть с хабами (коммутаторами, маршрутизаторами) иметь ячеистую структуру?

Ответ. Сеть с хабами и коммутаторами — нет. Сеть с маршрутизаторами — да.

22. Может ли сеть с хабами (коммутаторами, маршрутизаторами) иметь несколько доменов коллизий?

Ответ. Сеть с хабами — нет. Сеть с коммутаторами и маршрутизаторами — да.

23. Ethernet-сеть имеет ограничение на диаметр по причине недопущения поздних коллизий. Как, несмотря на это, построить длинную Ethernet-сеть?

Ответ. Разбить сеть на несколько доменов коллизий (при помощи коммутаторов или маршрутизаторов).

24. Расскажите алгоритм работы коммутатора.

Ответ. При включении питания коммутатор работает как обычный хаб: транслирует пакеты с одного порта на все другие. Но попутно коммутатор заносит в свою таблицу данные о соответствии адресов станций портам, с которых он получает пакеты. Таким образом, коммутатор заполняет таблицу маршрутизации и все больше изолирует порты друг от друга.

Рассмотрим подробнее алгоритм работы коммутатора на примере сети, изображенной на рис. 3.43.

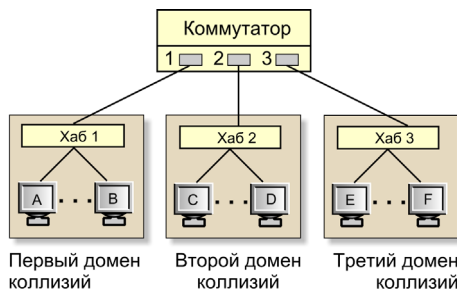


Рис. 3.43. Пример иерархической сети с коммутатором в корне

В начальный момент (при включении питания) таблица маршрутизации коммутатора пуста.

Пусть узел А передает пакет для узла В. Пакет содержит не только адрес получателя, но и адрес отправителя. Когда пакет приходит на порт 1, коммутатор делает в таблице первую запись:

Адрес узла	Порт, по которому находится узел
А	1

Теперь коммутатор ищет в таблице строчку для узла В, чтобы решить, что делать с пакетом: игнорировать, если В расположен на том же порту, что и А, или транслировать пакет в порт, к которому подключен В.

Строки с узлом В в таблице еще нет. Коммутатор вынужден работать как хаб: он транслирует пакет к неизвестному адресату на все порты, кроме того, с которого пакет получен, то есть на порты 2 и 3.

Пусть теперь узел F передает пакет для узла А.

В таблице появляется новая строка:

Адрес узла	Порт, по которому находится узел
А	1
F	3

Коммутатор находит в таблице порт получателя и передает пакет в порт 1.

Таким образом заполняется таблица маршрутизации, и коммутатор, начав как обычный хаб, быстро обучается, повышая свою «квалификацию».

25. В чем преимущество сетей с коммутацией пакетов перед сетями с разделяемыми средами?

Ответ. В среде с коммутацией пакетов:

- Можно использовать ячеистую структуру сети (многовариантность маршрутов). Это повышает надежность передачи: когда один маршрут выходит из строя, пакеты передвигаются по другому.

- Сообщения передаются быстрее: пакеты не транслируются по всем направлениям, а передвигаются по самому быстрому маршруту. Кроме того, не нужно перепосылать заново пакеты, испорченные коллизией (в сети нет коллизий).

- Так как пакеты не транслируются по всем направлениям, а передаются только получателю, повышается защита данных от несанкционированного использования.

26. В какой глобальной сети передача сообщений основана на принципе коммутации пакетов?

Ответ. В сети Интернет.

Вопросы зачета (с ответами)

В каждом задании Зачетного класса отметьте все правильные высказывания.

1. Разделяемая среда:

1.1. передача пакета повторяется через случайное время после коллизии

- 1.2. диаметр сети не ограничен
- 1.3. диаметр сети ограничен
- 1.4. сообщение разделяется на пакеты
- 1.5. пакеты транслируются на все станции сети
- 1.6. пакеты продвигаются шаг за шагом к станции назначения

Правильные ответы: 1, 3, 4, 5.

2. Среда с коммутацией пакетов:

- 2.1. передача пакета повторяется через случайное время после коллизии
- 2.2. диаметр сети не ограничен
- 2.3. диаметр сети ограничен
- 2.4. сообщение разделяется на пакеты
- 2.5. пакеты транслируются на все станции сети
- 2.6. пакеты продвигаются шаг за шагом к станции назначения

Правильные ответы: 2, 4, 6.

3. Устройства, сохраняющие единую разделяемую среду:

- 3.1. терминатор
- 3.2. трансивер
- 3.3. повторитель
- 3.4. хаб
- 3.5. коммутатор
- 3.6. маршрутизатор

Правильные ответы: 1, 2, 3, 4.

4. Устройства, разделяющие сеть на домены коллизий:

- 4.1. коммутатор
- 4.2. маршрутизатор
- 4.3. терминатор
- 4.4. трансивер
- 4.5. повторитель
- 4.6. хаб

Правильные ответы: 1, 2.

5. Диаметр сети Ethernet:

- 5.1. максимальный отрезок кабеля, соединяющий две соседние станции
- 5.2. максимальная суммарная длина кабеля, соединяющего две станции
- 5.3. суммарная длина кабеля, соединяющего все станции
- 5.4. минимальная длина кабеля, соединяющего две станции

Правильный ответ: 2.

6. Каковы причины ограничения диаметра сети Ethernet:

- 6.1. не допускать в среде поздние коллизии
- 6.2. не допускать в среде ранние коллизии
- 6.3. не допускать в среде коллизии
- 6.4. не допускать монополизацию среды

Правильный ответ: 1.

7. Что влияет на вычисление допустимого диаметра сети Ethernet:

- 7.1. скорость передачи данных в среде
- 7.2. скорости распространения сигнала в среде
- 7.3. минимальная длина пакета
- 7.4. максимальная длина пакета
- 7.5. число станций в сети

Правильные ответы: 1, 2, 3.

8. Чем коммутатор отличается от хаба:

- 8.1. у коммутатора больше портов
- 8.2. у коммутатора есть программное обеспечение, хаба — нет
- 8.3. коммутатор маршрутизирует пакеты, а хаб — нет
- 8.4. коммутатор делит сеть на домены коллизий, а хаб — нет
- 8.5. коммутатор проверяет контрольную сумму пакета, а хаб — нет

Правильные ответы: 2, 3, 4.

9. Сколько в сети на рис. 3.44 доменов коллизий?



Рис. 3.44. Сколько здесь доменов коллизий?

Правильный ответ: 1.

10. Сколько в сети на рис. 3.45 доменов коллизий?

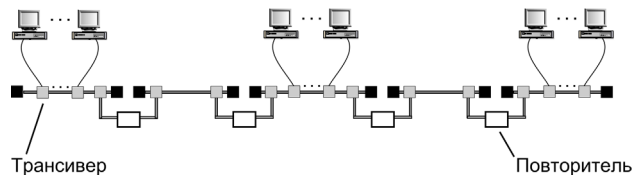


Рис. 3.45. Сколько здесь доменов коллизий?

Правильный ответ: 1.

11. Сколько в сети на рис. 3.46 доменов коллизий?

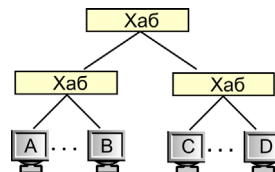


Рис. 3.46. Сколько здесь доменов коллизий?

Правильный ответ: 1.

12. Сколько в сети на рис. 3.47 доменов коллизий?

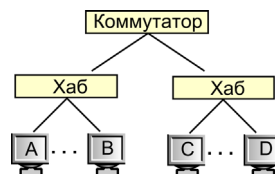


Рис. 3.47. Сколько здесь доменов коллизий?

Правильный ответ: 2.

13. Сколько в сети на рис. 3.48 доменов коллизий?

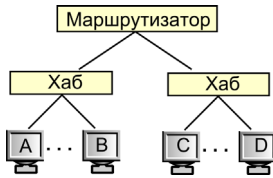


Рис. 3.48. Сколько здесь доменов коллизий?

Правильный ответ: 2.

14. Сколько в сети на рис. 3.49 доменов коллизий?

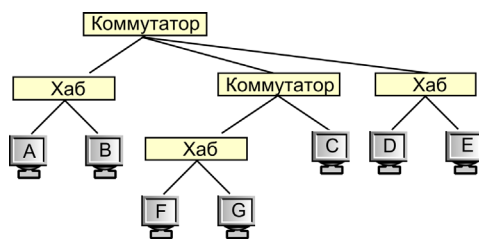


Рис. 3.49. Сколько здесь доменов коллизий?

Правильный ответ: 4.

FAQ

Приводятся ответы на следующие вопросы:

- Что такое FAQ?
- Хаб — это сетевой адаптер?
- В чем разница между аналоговыми и цифровыми модемами?
- Что такое проху-сервер?
- Как поработать в Фидо?
- Для чего перекручивают провода в кабеле “витая пара”?

1. Что такое FAQ?

Ответ. FAQ (эф эй кью) — это сокращение от *Frequently Asked Questions* (часто задаваемые вопросы). Иногда этот раздел (содержащий ответы на часто задаваемые вопросы по некоторой теме) называют ЧАВО — **Ч**Асто задаваемые **В**опросы. Последняя аббревиатура навеяна, конечно, названием института из повести Аркадия и Бориса Стругацких “Понедельник начинается в субботу”: НИИЧАВО — Научно-Исследовательский Институт ЧАродейства и ВОлшебства.

“НИИЧАВО, — подумал я. — Научно-исследовательский институт... Чаво? В смысле — чего? Чрезвычайно Автоматизированной Вооруженной Охраны? Черных Ассоциаций Восточной Океании?”

Аркадий Стругацкий, Борис Стругацкий

2. Хаб — это сетевой адаптер?

Ответ. Нет. Сетевой адаптер — это преобразователь сигналов между компьютером и сетью. Хаб используется для соединения сегментов кабеля в сети.

Соединение хаба с компьютером выполняется через сетевой адаптер.

3. В чем разница между аналоговыми и цифровыми модемами?

Ответ. Аналоговый модем преобразует компьютерные сигналы в звуковые (до 4 КГц) и передает их по обычным телефонным сетям. Можно услышать, как во время передачи такой модем жужжит и шипит.

Аналоговые модемы широко распространены благодаря сравнительно низкой стоимости и простоте подключения.

Максимальная скорость передачи данных с помощью аналогового модема ограничена 56 Кбит/с.

Во время соединения по аналоговому модему телефон становится недоступным для обычного разговора.

Цифровые модемы используют для передачи данных на частоте, гораздо выше звуковых частот (от 4 КГц до 1–2 МГц), что позволяет достигать скоростей передачи данных до нескольких Мбит/с.

Так как низкие частоты не используются, то это позволяет вести телефонный разговор, не прерывая соединение по цифровому модему.

Для работы с цифровыми модемами на АТС (Автоматической Телефонной Станции) должно быть установлено специальное оборудование. В этом надо убедиться при покупке цифрового модема.

Для разделения сигналов телефона и модема обычно приходится устанавливать дополнительный частотный делитель: *сплиттер*, или *частотный микрофильтр*. В некоторых случаях это требует переделки телефонной проводки.

В настоящее время цифровые модемы работают по нескольким стандартам: ADSL, VDSL, SHDSL и др. Эти технологии имеют общее название xDSL (DSL расшифровывается как *Digital Subscriber Line* — цифровая абонентская линия).

4. Что такое проху-сервер?

Ответ. Проху-сервер (пишут и по-русски — “прокси-сервер”) — это программное обеспечение на сервере, которое выступает посредником между клиентом сервера и Интернетом.

Смысл посредничества: сохранять (накоплять) запрашиваемую из Интернета информацию на локальном диске.

При каждом запросе клиента прокси-сервер сначала ищет информацию у себя на диске, и только если там ее нет, начинает работать с Интернетом.

Прокси-сервер ускоряет обслуживание клиента и удешевляет ее: информация с локального диска поставляется, как правило, даром.

5. Как поработать в Фидо?

Ответ. В Фидо существует строгая иерархия пользователей: сетевой координатор, хаб, нод, поинт.

- **Поинт.** Самый низший уровень. Может пользоваться почтовой сетью, конференциями (группы новостей), запрашивать файлы с любой станции Фидо.

- **Нод (или нода).** Аналог провайдера в Интернете. Нод, к которому прикреплен поинт, является для этого поинта *боссом*.

- **Хаб.** Координатор, курирующий большую группу нодов.

- **Сетевой координатор.** Курирует работу подсети Фидо: города, региона, страны, континента. Сетевые координаторы также связаны между собой иерархической зависимостью.

Для подключения к сети Фидо нужно найти “фидошника”, который рекомендовал бы вас своему боссу и помог бы настроить программы для работы с сетью Фидо.

6. Для чего перекручивают провода в кабеле “витая пара”?

Ответ. Скручивание проводов снижает влияние помех на сигналы, передаваемые по такому кабелю.

Сигнал с пары проводов считывается, как разность потенциалов (напряжений) на них.

Помеха создает на проводах добавочные потенциалы, но так как провода в витой паре равномерно перекручены, на каждом из них возникают примерно одинаковые добавочные напряжения, и они взаимно гасятся при вычитании в приемном устройстве.

Пусть передаваемый по проводам сигнал создает на них потенциалы в 6 и 4 вольта. Принимающая станция считывает разность потенциалов 2 вольта при передаче без помех (рис. 3.50).

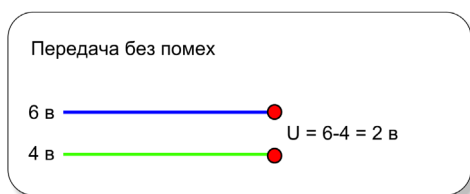


Рис. 3.50. Передача без помех

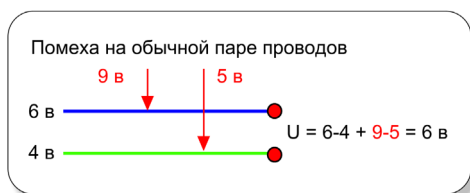


Рис. 3.51. Передача с помехой по нескрученной паре проводов

На проводах обычного кабеля помеха создает разные дополнительные потенциалы в зависимости от их удаленности от источника помехи (рис. 3.51). В итоге приемник фиксирует разность потенциалов на 4 вольта больше (сигнал искажен).

Так как провода витой пары перекручены (расположены одинаково от помехи), паразитный потенциал на них получается одинаковым, и в итоге сигнал не искажается (рис. 3.52).

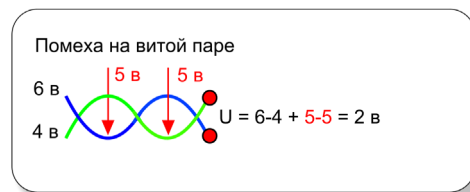


Рис. 3.52. Передача с помехой по витой паре

Полезные ссылки

1. Компьютерные сети от простого к сложному network.xsp.ru

Описание с сайта. Данный сайт создан для того, чтобы рассказать вам, что такое сеть, как ее построить и как обслуживать. Я попытаюсь объяснить, что вам для этого потребуется, какие имеются подходы к построению сети. Здесь приведены различные примеры сетей: от небольших (2 компьютера) до сетей масштаба предприятия. Все, что здесь представлено, взято из моего личного опыта по построению и администрированию сетей.

2. История Интернета в России www.nethistory.ru

Описание с сайта. “История Интернета в России” — это исследовательский проект, целью которого является сбор и анализ информации о развитии интернет-технологий в России. Периодически на сайте публикуются интервью, анонсируются новые статьи и книги об Интернете.

3. Планета Интернет www.netplanet.ru

Описание с сайта. “Планета Интернет” является первым в России периодическим иллюстрированным изданием, посвященным международной сети Интернет. Журнал освещает события, происходящие в Сети и вокруг нее, рассматривает современные технологии, оказывающие воздействие на жизнь и культуру современного информационного общества.

Разделы и рубрики журнала: *Тема номера, Новости, Интернет и бизнес, Закладки, Персона, Вопрос-ответ, Обзорение, Хакеры, На досуге, Спортнет, Технологии, Расследование.*



Издательство «Просвещение»

Весь мир будет твой!

ИНФОРМАТИКА

УМК «Информатика 5» и «Информатика. Алгоритмика» для 6–7-х классов
авторов **А.Л. Семенова и др.**

«Информатика, 5»

Курс «Информатика, 5» авторов А.Л. Семенова и Т.А. Рудченко является продолжением и развитием курса «Информатика 2–4» тех же авторов. При этом работа с курсом в 5-м классе не предполагает обязательного предварительного изучения курса «Информатика, 2–4». Материал курса «Информатика, 5» подобран таким образом, чтобы он был интересен как детям, прошедшим в начальной школе курс информатики, так и детям, впервые начинающим изучение этой дисциплины в 5-м классе. В курсе начальной школы вся теоретическая и практическая деятельность детей проходила на материале графических и телесных объектов. При изучении курса «Информатика, 5» работа на графическом и телесном уровне также является основной. Все понятия курса, в том числе и абстрактные, иллюстрируются в конкретных (телесных и графических) моделях, доступных для понимания детей. Это многократно увеличивает качество усвоения содержания курса. Однако в 5-м классе увеличивается объем учебных текстов. Если в курсе 2–4-х классов текст в основном имел место как пояснение к картинкам, то теперь назначение текста становится более разнообразным. В том случае, если понятие можно определить словами кратко и ясно, появляются формальные определения. Также появляются краткие описательные тексты, поясняющие новое понятие или содержащие примеры. Везде, где это возможно, тексты по-прежнему сопровождаются графической иллюстрацией. Таким образом, авторы стремятся сделать переход от телесно-графического восприятия материала в начальной школе к абстрактному — в старшем звене, максимально безболезненным и постепенным.

В УМК «Информатика, 5» входят:

- учебник,
- тетрадь проектов,
- методическое пособие для учителя.

Учебник

Учебник содержит страницы, где даются определения понятий — *листы определений*, и *задания*. Страницы с заданиями содержат задачи по темам, представленным на листах определений. Разумеется, блоки задач — всегда однородны — часто встречаются задачи на повторение и обобщение, многие задачи — комплексные, использующие большинство уже пройденных тем.

Тетрадь проектов

В тетради проектов собраны материалы, которые потребуются при проведении специальных проектных уроков: задания для детей, карточки со словами, рабочие тексты и прочие раздаточные материалы. Тетрадь проектов представляет собой комплект, состоящий из собственно сборника проектов, раздаточного материала и вкладыша, содержащего космический снимок участка Москвы и схему того же участка. На другом вкладыше тетради проектов помещены варианты контрольных работ. Раздаточный материал содержит заготовки таблиц, игровых полей и пр., которые понадобятся учащимся при решении задач из учебника. Вырезание и наклеивание в тетрадь таких заготовок поможет учащимся существенно сократить время на техническую подготовку к решению задачи.

Книга для учителя

Книга для учителя содержит, помимо программы, описания курса и общих советов по проведению занятий, обсуждения каждой новой темы и блока задач, относящегося к этой теме, а также тексты контрольных работ. В отдельной главе пособия приведены подробные описания работы в проектах.

«Информатика. Алгоритмика» для 6–7-х классов

Основная цель курса «Информатика. Алгоритмика» для 6–7-х классов — формирование у школьников основ алгоритмического мышления. Под способностью алгоритмически мыслить понимается умение решать задачи различного происхождения, требующие составления плана действий для достижения желаемого результата. Алгоритмическое мышление является неотъемлемой частью научного взгляда на мир. В то же время умение планировать свои и чужие действия и предсказывать их последствия необходимо в повседневной практической деятельности современного человека, чем бы он ни занимался. Курс алгоритмики тесно связан как с другими частями курса математики (доказательство утверждений, начала теории чисел, системы координат, геометрические фигуры, построение графиков), так и с другими школьными курсами, в первую очередь курсом физики. Курс «Информатика. Алгоритмика» для 6–7-х классов рассчитан на обучение в течение двух лет учащихся 6–7-х классов средней общеобразовательной школы. Бескомпьютерный вариант предусматривает проведение одного урока в неделю. При наличии доступа к компьютерам, на которых установлена система программной поддержки «Алгоритмика», целесообразно увеличить нагрузку за счет введения дополнительно одного академического часа компьютерных занятий раз в две недели.



При этом объем изучаемого материала не увеличивается; упор делается на отработку знаний и навыков составления алгоритмов.

Объем предварительных знаний, необходимых для эффективного освоения алгоритмики, невелик: достаточно умения свободно выполнять арифметические операции над целыми числами. Ее предметом служат комбинаторные объекты, которые легко описываются; с ними можно работать руками, а доказательства производить методом полного перебора. Познание может происходить при активном использовании игр, театрализации, элементов соревнования.

Первый год работы предназначен для первоначального знакомства с построением и анализом алгоритмов. Вместе с тем уже на этом этапе вводятся все основные понятия: алгоритм, процедура, условия, эффективность, рекурсия и др. Используемый алгоритмический язык не вводится изначально, а строится постепенно, по мере появления необходимости в тех или иных конструкциях. Большое внимание уделяется основам логики — условиям и их использованию. Исполнители, используемые в курсе, традиционны. Исключение составляет введенный А.К. Звонкиным Исполнитель — Директор строительства. Это одна из первых попыток познакомить учащихся с понятием параллельного программирования. Знакомство происходит на простом и в то же время очень содержательном материале строительных кубиков. Единжды введенные исполнители в дальнейшем активно используются на протяжении всего курса.

Второй год работы посвящен освоению переменных. В первой части курса рассматривается использование переменных во введенных ранее исполнителях и обсуждаются достигаемые за счет этого преимущества и упрощения. Затем достигнутые преимущества развиваются за счет введения новых исполнителей. Графические исполнители первого года обучения применяются для выработки общих методов рисования различных графиков. Здесь же анализируется использование массивов. Знакомство с эффективными вычислениями осуществляется на примерах алгоритма Эвклида, перевода числа из одной позиционной системы счисления в другую и схемы Горнера.

В УМК «Информатика, 6. Алгоритмика»

(авторы А.К. Звонкин, С.К. Ландо, А.Л. Семенов) входят:

- учебник,
- книга для учителя.

Учебник и задачник

Книга состоит из введения, одиннадцати глав теоретического материала и задачника. Введение содержит сведения о необходимости развития алгоритмического мышления как искусства размышлять, умения планировать свои действия, способности предусматривать различные обстоятельства и поступать в соответствии с ними. В первой, теоретической части учебника рассматриваются способы решения задач с использованием программ, выполняемых некоторыми исполнителями. В качестве исполнителей в учебнике используются Черепашка,

Чертежник, Робот, Кузнечик. Материалы учебника развивают способность разбиения сложных задач на подзадачи и решения задач последовательно, описывая некоторый алгоритм с помощью созданного авторами учебного языка программирования. Во второй части учебника собраны и систематизированы задачи. Учебник соответствует федеральному учебному плану по математике и информатике и рассчитан на безмашинное обучение (существует и распространяется программное обучение учебника).

Книга для учителя

Состоит из введения и шести глав. Методическое пособие предназначено для учителей, работающих по учебнику «Информатика, 6» и состоит из двух частей. Первая часть содержит методические рекомендации, примерное планирование материала курса, варианты поурочных планов и решение упражнений из учебника. Во второй части пособия даются решения всех задач из задачника.

В УМК «Информатика, 7. Алгоритмика»

(авторы М.Н. Вялый, С.К. Ландо, А.Л. Семенов) входят:

- учебник,
- книга для учителя.

Учебник и задачник (готовится к изданию)

В этой части курса рассматривается использование переменных во введенных ранее исполнителях и обсуждаются достигаемые за счет этого преимущества и упрощения. Затем достигнутые преимущества развиваются за счет введения новых исполнителей. Графические исполнители первого года обучения применяются для выработки общих методов рисования различных графиков. Здесь же анализируется использование массивов. Знакомство с эффективными вычислениями осуществляется на примерах алгоритма Эвклида, перевода числа из одной позиционной системы счисления в другую и схемы Горнера.

Учебник соответствует федеральному учебному плану по математике и информатике и рассчитан на безмашинное обучение (существует и распространяется программное обучение учебника).

Значительная часть книги посвящена изучению вопросов, играющих принципиальную роль в современной жизни, однако обычно не рассматривающихся в школе. К этим вопросам относятся: случайность и ее применение, шифрование и дешифровка, игровые алгоритмы. Первоначальное знакомство с этими областями человеческого знания может быть проведено на доступном и привлекательном для школьников материале.



Издательство «Просвещение»
127521, Москва,
3-й проезд Марьиной Рощи, 41
Тел.: (495) 789-3040
Факс: (495) 789-3041
E-mail: prosv@prosv.ru
http://www.prosv.ru

Служба «Книга — почтой»
Заказ и отправка книг по почте
102001, Москва,
а/я «Почтовый Торговый Дом»
Тел.: (495) 540-6061
E-mail: prosv@post.ru, zakaz@ptdom.ru
http://www.ptdom.ru

Фирменные магазины издательства «Просвещение»
115304, Москва,
ул. Луганская, д. 7, корп. 2
Тел.: (495) 322-2822

119311, Москва,
пр-т Вернадского, 11/19
Тел.: (495) 930-5050
Тел./факс: (495) 930-5040
E-mail: mag-info@prosv.ru

7 ноября 2006 года

ПРОГРАММА ДНЯ**ПРЕДМЕТЫ
ЕСТЕСТВЕННО-НАУЧНОГО ЦИКЛА**в Московском городском доме учителя: Москва, ул. Пушечная,
д. 4, стр. 2 (ст. м. «Кузнецкий мост»)

**ВСЬ ДЕНЬ РАБОТАЕТ ВЫСТАВКА-ПРОДАЖА УЧЕБНО-МЕТОДИЧЕСКОЙ ЛИТЕРАТУРЫ ВЕДУЩИХ ИЗДАТЕЛЬСТВ
БОЛЕЕ 1000 НАИМЕНОВАНИЙ КНИГ ДЛЯ УЧИТЕЛЯ ПРЕДСТАВЛЯЮТ ИЗДАТЕЛЬСТВА**

«Абрис», «Айрис», «АСТ-Пресс Образование», «Владос», «Дрофа», «Илекса», «Интеллект-Центр», «КУДИЦ-ПРЕСС», «МЕДИАХАУС», «Мнемозина», «Просвещение», «СГУ», «СМИО Пресс», «Физматлит»

	АУД. № 2–14	АУД. № 0–1	ПАРКЕТНЫЙ ЗАЛ
11.00	ДРОФА «Геоэкология и страноведение: синтез в учебной географии» О.А. Климанова, к.геогр.н., ст. научный сотрудник кафедры физической географии мира и геоэкологии МГУ им. М.В. Ломоносова, автор учебников и методических пособий «Новый проект издательства «Дрофа»: интерактивные пособия по географии» О.А. Панасенкова, зав. редакцией географии изд-ва «Дрофа»	ИЛЕКСА «Книги по физике нового поколения издательства «Илекса» Л.Э. Генденштейн, к.ф.-м.н., учитель-методист, научный сотрудник ИСМО РАО Л.А. Кирик, отличник просвещения СССР, учитель-методист, научный сотрудник ИСМО РАО	ПРОСВЕЩЕНИЕ «Обзор новой учебной литературы по географии издательства «Просвещение»
12.30	АСТ-ПРЕСС ШКОЛА «Экология и устойчивое развитие» Н.М. Мамедов, автор УМК «Экология»	ИЛЕКСА «Деятельностный подход в преподавании математики» А.П. Ершова, учитель-методист, отличник образования Украины, Соросовский учитель В.В. Голобородько, учитель высшей категории	ПРОСВЕЩЕНИЕ «Научно-методические основы курса физики для 7–9-х классов (УМК «Ломоносов»). Итоговая аттестация учащихся по физике в основной школе» А.А. Фадеева, д.п.н., профессор, гл. научный сотрудник ИСМО РАО
14.00	АЙРИС «Эффективные методы решения неравенств, содержащих логарифмы, радикалы и модули. Авторская методика» С.И. Колесникова, преп. математики физико-технического класса шк. № 463 г. Москвы, с 1962 г. ст. преп. кафедры высшей математики МФТИ, ведущий специалист ЗФТШ при МФТИ, лектор курсов повышения квалификации при МФТИ	АСТ-ПРЕСС ШКОЛА «Универсальные учебные атласы по географии как важнейшая составляющая современного географического образования» С.П. Потеряйко, методист	ПРОСВЕЩЕНИЕ «Учебно-методический комплект М.И. Башмакова в профильном обучении» М.И. Башмаков, д.ф.-м.н., профессор, академик РАО
15.30			ПРОСВЕЩЕНИЕ «Курс информатики А.Л. Семёнова и др. во 2–6-х классах» А.Л. Семёнов, д.ф.-м.н., профессор, лауреат премии Президента РФ в области образования, заслуж. работник высшей школы, член-корреспондент РАО, ректор Московского института открытого образования

Дополнительную информацию о фестивале можно найти на сайте ИД «Первое сентября» www.1september.ru или получить по телефонам редакции: (495) 249-31-38, 249-98-70.

ВХОД СВОБОДНЫЙ

В мир информатики

79 (18–31 октября)

Газета для пытливых учеников
и их талантливых учителей

Рекуррентные соотношения в Microsoft Excel

В нашей газете публиковались материалы (см. [1]), посвященные последовательностям чисел, построенным на основе так называемых “рекуррентных соотношений” — формул, выражающих очередной член последовательности через один или несколько предыдущих членов. Например, для арифметической прогрессии справедлива такая формула: $a_i = a_{i-1} + d$ (каждый последующий член равен предыдущему, увеличенному на разность прогрессии). В [1] были рассмотрены вопросы, связанные с вычислением n -го члена последовательности, заданной рекуррентным соотношением, средствами программирования. А можно ли решать подобные задачи в электронной таблице Microsoft Excel? Можно, и делается это следующим образом.

Если для расчета n -го члена последовательности есть формула (например, для арифметической прогрессии она имеет вид: $a_n = a_1 + d(n - 1)$), то задача решается довольно просто:

	A	B	C	D
1	Расчет n-го члена арифметической прогрессии			
2	Задайте первый член прогрессии:			
3	Задайте разность прогрессии:			
4	Задайте номер искомого члена прогрессии:			
5	Искомое значение равно:			
6				

Формула в ячейке B5: = B2+B3*(B4-1).

Чаще, однако, такой формулы нет или она неизвестна. В этом случае члены последовательности вычисляются по рекуррентному соотношению один за другим от $i = 1$ до $i = n$. Рассмотрим задачу: “Последовательность чисел 2, 3, 5, 9, 17, ... образуется по закону: $a_1 = 2$; $a_k = 2 * a_{k-1} - 1$, ($k = 2, 3, \dots$). Определить a_{50} ”.

Здесь решение можно оформить следующим образом¹:

	A	B	C
1	k	a	
2	1		2
3	2		3
4	3		5
5	4		9
...			
50	49	281 474 976 710 657	
51	50	562 949 953 421 313	
52			

Для расчетов в ячейку B3 вводится формула: =B2*2-1, которая затем распространяется (копируется) на ячейки диапазона B4:B51 (в ячейке B2 записывается известное значение 2). В ячейках с числами в столбце B установлен так называемый “числовой формат значений с разделителем в виде пробела между тройками чисел”².

Однако такое оформление листа нерационально — для определения единственного значения используются 50 (!) ячеек (без учета ячеек в столбце A). Можно значительно сократить размеры используемой части листа.

1. Для начала запишем на лист следующие (вспомогательные) данные:

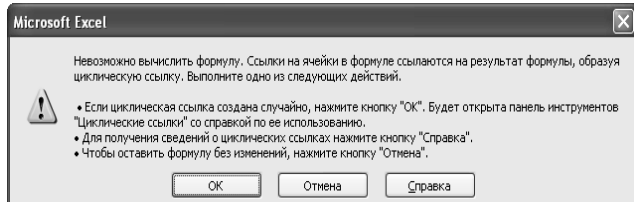
	A	B	C
1	Для начала расчетов введите любой текст ->		
2	Искомое значение =		
3			

¹ При необходимости можно также оформить лист таким образом, чтобы можно было получать значение любого члена последовательности (см. [2]).

² Для установки такого формата следует использовать пункт меню **Формат**, подпункт **Ячейки**, вкладку **Число** и в разделе **Форматы** выбрать из списка **Числовые форматы** строку **Числовой**, указав при этом в поле **Число десятичных знаков** значение 0 и поставив “галочку” рядом с надписью **Разделитель групп разрядов** ().

В ячейке В2 попытаемся получить искомый результат путем использования всех “предыдущих” 49 значений. Смысл текста в ячейке В1 раскроем чуть позже.

2. В ячейку В2 введем формулу =B2*2-1 и ... — да, конечно, сразу же появится сообщение об ошибке:

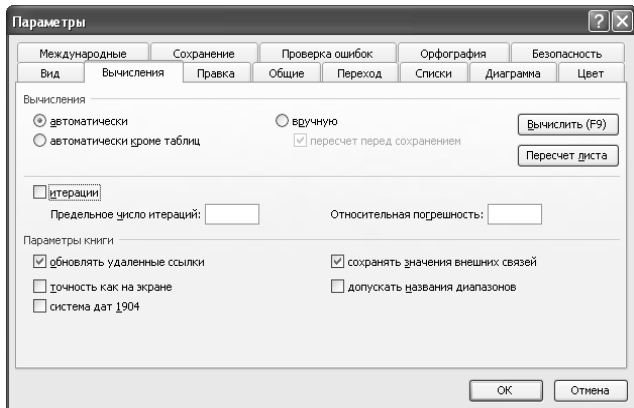


Смысл этого сообщения в том, что в заданной формуле имеется так называемая “циклическая ссылка” — результат расчетов в ячейке В2 зависит от значения в этой же ячейке. И хотя нам нужно именно это — ведь в этом по определению заключается смысл рекуррентных формул, Microsoft Excel такого не допускает (пока!).

В появившемся окне с сообщением об ошибке щелкаем на кнопке **Отмена**, а затем очищаем ячейку В2.

3. Для того чтобы можно было использовать формулы с циклической ссылкой (рекуррентным соотношением), следует:

— в пункте меню **Сервис** выбрать подпункт **Параметры** и в появившемся окне — вкладку **Вычисления**:



— поставить “галочку” (✓) рядом с надписью итерации, а в поле **Предельное число итераций** записать значение 49 (напомним, что нам надо определить 50-й член последовательности), после чего щелкнуть на кнопке **ОК**.

4. В ячейке В2 запишем такую формулу: =ЕСЛИ(В1""; 1; В2*2-1) — и теперь Microsoft Excel “промолчит” (☺). Смысл этой формулы достаточно ясен: пока ячейка В1 пустая, в ячейке В2 должно отражаться начальное значение, равное 1, а после заполнения ячейки В1 в ячейке В2 начнут выполняться расчеты по рекуррентному соотношению (расчеты по заданной формуле будут проведены 49 раз).

5. В результате, если все оформлено правильно, на листе появится искомое значение:

	А	В	С
1	Для начала расчетов введите любой текст ->	д	
2	Искомое значение =	562 949 953 421 313	
3			

Эффектно, не правда ли?

При необходимости можно получить не один, а несколько членов последовательности. Для этого следует в поле **Предельное число итераций** (см. рисунок выше) задать значение 1. В этом случае после ввода текста в ячейку В1 в ячейке В2 появится значение второго члена последовательности — 3. Нажимая затем функциональную клавишу **F9**, можно получать в ячейке третий, четвертый и другие члены последовательности. Так удобно проводить расчеты, когда искомым значениям немного. Если же нужно получить одно число, а итераций (расчетов по рекуррентной формуле) много, то лучше установить необходимое значение числа итераций в окне **Параметры**, вызвав его с помощью меню, как это описано выше.

Задания для самостоятельной работы

1. Подготовить лист для расчета 100-го члена последовательности, заданной рекуррентным соотношением $a_k = a_{k-1} + 1/k$ ($k = 2, 3, \dots$) при $a_1 = 1$.

2. Получить первые 15 членов последовательности 2, 5, 14, 41, ... (закон построения последовательности установите самостоятельно).

3. Начав тренировки, лыжник в первый день пробежал 10 км. Каждый следующий день он увеличивал пробег на 10% от пробега предыдущего дня. Определить, какой суммарный путь он пробежал за первые 7 дней тренировок.

4. В некотором году (назовем его условно первым) на участке в 100 гектар средняя урожайность ячменя составила 20 центнеров с гектара. После этого каждый год площадь участка увеличивалась на 5%, а средняя урожайность на 2%. Определить, какой урожай будет собран за первые шесть лет.

5. Найти 10-й член последовательности, начинающейся с числа 2,5, в которой каждый следующий член равен сумме обратных величин всех предыдущих.

6. Найти 10-й член последовательности, начинающейся с числа 2,5, в которой каждый следующий член равен обратной сумме всех предыдущих членов.

Ответы присылайте в редакцию.

Литература

1. Школа программирования: Тематический выпуск газеты-вкладки “В мир информатики” (“Информатика” № 11/2005).

2. Обработка переменного количества исходных величин в Microsoft Excel. / “В мир информатики” № 8 (“Информатика” № 40/2003).

Продолжение — в следующем выпуске

Школа программирования

Основы программирования на Visual Basic

Продолжение. Начало см. “В мир информатики” № 69–73, 78 (“Информатика” № 4–8, 19/2006)

Н.М. Тимофеева, г. Обнинск Калужской обл.

Пример программирования № 6 “Бросание игральных костей”

1. Постановка задачи

Разработайте программу, которая будет имитировать бросание пары игральных костей.

План решения

На рис. 1 показан предлагаемый дизайн экрана:

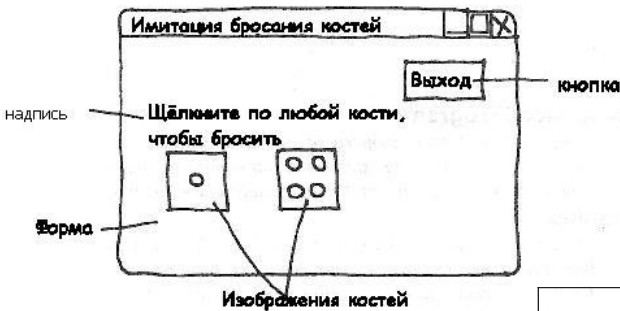


Рис. 1

Назначение каждого элемента управления:

- 1) элементы управления **Image (Рисунок)** — для графического представления игральных костей;
- 2) надпись — для вывода подсказки пользователю;
- 3) кнопка — для завершения программы по щелчку на ней.

По щелчку на любом элементе управления **Image (Рисунок)** на обоих элементах должно появляться изображение “выпавшей” кости. В N–S-диаграмме для процедур, реагирующих на событие “Щелчок на элементе управления **Image**”, используем блоки множественного выбора [1] (см. диаграммы справа).

Видно, что мы имеем ситуацию, когда целесообразно использовать так называемую “общую процедуру” для двух элементов³.

³ Об общих процедурах см. предыдущую статью. — Ред.

Создание интерфейса

Перед началом работы над программой в Visual Basic вам нужно воспользоваться графическим редактором Paint или другим, чтобы создать изображения игральных костей. Вам понадобятся шесть отдельных *bmp*-файлов, каждый из которых будет картинкой одной из граней кубика (кости). Совет: просто нарисуйте квадрат 2 × 2 см с черными точками на нем. Сохраните файлы с этими картинками в папке ПРИМЕР6 с именами *1.bmp*, *2.bmp* и т.д.

Запустите Visual Basic, начните новый проект и разместите на форме два элемента управления **Image**, одну надпись и одну кнопку (см. рис. 2):

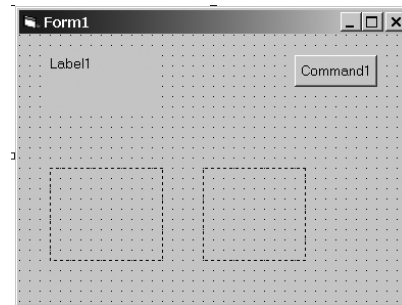


Рис. 2

Получить случайное число от 1 до 6						
Выбрать выпавший номер						
1	2	3	4	5	6	
Загрузить картинку 1.bmp в первое изображение	Загрузить картинку 2.bmp в первое изображение	Загрузить картинку 6.bmp в первое изображение	

Получить случайное число от 1 до 6						
Выбрать выпавший номер						
1	2	3	4	5	6	
Загрузить картинку 1.bmp во второе изображение	Загрузить картинку 2.bmp во второе изображение	Загрузить картинку 6.bmp во второе изображение	

Установка свойства

Установите следующие свойства формы и элементов управления:

Form1

<i>Caption</i>	Имитация бросания костей
----------------	--------------------------

Label1

<i>Caption</i>	Щелкните по любой кости, чтобы бросить
----------------	--

Image1

<i>Name</i>	ImgКость1
-------------	-----------

Image2

<i>Name</i>	ImgКость2
-------------	-----------

Command1

<i>Name</i>	cmdВыход
<i>Caption</i>	Выход

Выберите свойство `Picture` каждого элемента управления **Image**, щелкните на многоточии (...), в появившемся диалоговом окне из созданных вами графических файлов выберите файл, который будет начальным для соответствующей игральной кости. Вместо щелчка по многоточию для вызова диалогового окна можно просто дважды щелкнуть по самому названию свойства `Picture`.

Написание кода

Как отмечалось выше, так как щелчок по любой из игральных костей будет имитировать бросание обеих костей, то код для каждой кости будет одним и тем же. Это означает, что мы имеем ситуацию, для которой подходит общая процедура.

Перейдите в окно кода и с помощью меню **Tools | Add Procedure** добавьте общую процедуру с именем `БросатьКости`:

```
Private Sub БросатьКости()
    Dim Кость1 As Integer, Кость2 As Integer
    Randomize
    'Бросить первую кость
    Кость1 = Int(Rnd * 6) + 1
    'Выбрать подходящий графический файл
    Select Case Кость1
    Case 1
        ImgКость1.Picture = LoadPicture("1.bmp")
    Case 2
        ImgКость1.Picture = LoadPicture("2.bmp")
    Case 3
        ImgКость1.Picture = LoadPicture("3.bmp")
    Case 4
        ImgКость1.Picture = LoadPicture("4.bmp")
    Case 5
        ImgКость1.Picture = LoadPicture("5.bmp")
    Case 6
        ImgКость1.Picture = LoadPicture("6.bmp")
    End Select
    'Бросить вторую кость
    Кость2 = Int(Rnd * 6) + 1
    Select Case Кость2
    'Выбрать подходящий графический файл
```

```
Case 1
```

```
Img Кость2.Picture = LoadPicture("1.bmp")
```

```
Case 2
```

```
Img Кость2.Picture = LoadPicture("2.bmp")
```

```
Case 3
```

```
Img Кость2.Picture = LoadPicture("3.bmp")
```

```
Case 4
```

```
Img Кость2.Picture = LoadPicture("4.bmp")
```

```
Case 5
```

```
Img Кость2.Picture = LoadPicture("5.bmp")
```

```
Case 6
```

```
Img Кость2.Picture = LoadPicture("6.bmp")
```

```
End Select
```

```
End Sub
```

В результате код для элементов управления **Image** становится очень коротким:

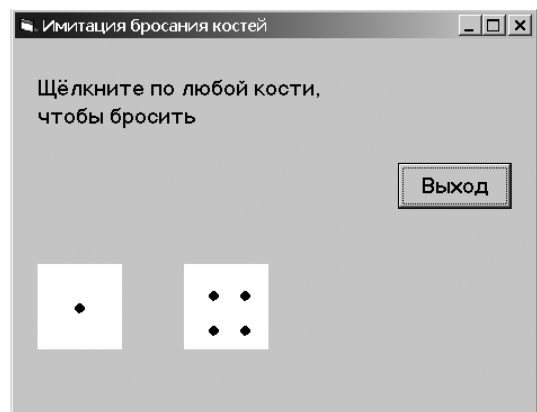
```
Private Sub ImgКость1_Click()
    'Вызываем процедуру БросатьКости
    БросатьКости
End Sub
Private Sub ImgКость2_Click()
    'Вызываем процедуру БросатьКости
    БросатьКости
End Sub
```

Сохранение проекта

Создайте папку *ПРИМЕР6* и сохраните в ней форму и проект под именами *кости.frm* и *кости.vbp*.

Тестирование приложения

Запустите программу.



Проверьте, что изображения появляются при щелчке на каждом элементе управления.

Взгляд назад

Убедитесь, что программа выполняет все, что требуется. Может ли так произойти, что одна или обе кости не будут "брошены"? Как это может быть преодолено?

Пояснения

1. В этом примере программирования вы познакомились с элементом управления **Image** (Рисунок), который во многом похож на использовавшийся ранее элемент управления **PictureBox**. Преимущества элемента **Image** перед элементом **PictureBox** в том, что он требует меньше системных ресурсов. Однако у элемента **Image** меньше возможностей (список свойств

короче). При использовании графики вам придется решать, какой элемент управления больше подходит к конкретной ситуации.

2. Это наша (ваша) первая программа, которая, кроме значения свойств элементов управления, использует переменные величины (в предыдущем проекте применялись постоянные величины — константы). Более подробно о переменных мы поговорим в следующем выпуске. Здесь же заметим, что они объявляются при помощи оператора Dim:

```
Dim Кость1 As Integer, Кость2 As Integer
```

При объявлении указывается имя переменной и ее тип (в нашем случае объявлены две переменные, которые будут хранить целые значения). Так же, как и для констант, имена переменных придумывает сам программист (желательно, чтобы имя говорило о “назначении” переменной величины).

3. В программе генерируются случайные числа (с помощью функции Rnd), но для того, чтобы гарантировать, что каждый раз не будет генерироваться одна и та же последовательность случайных чисел, используется оператор Randomize.

4. Функция Rnd возвращает случайное вещественное число, которое больше нуля, но меньше единицы (тип данных Single), функция Int удаляет дробную часть числа, превращая его в целое:

— $Rnd * 6$ — будет выдавать вещественное число, большее или равное 0 и меньшее 6;

— $Int(Rnd * 6)$ — будет возвращать одно из целых чисел: 0, 1, 2, 3, 4, 5;

— $Int(Rnd * 6) + 1$ — будет выдавать одно из целых чисел: 1, 2, 3, 4, 5, 6.

5. В этой программе впервые используется оператор варианта (выбора) Select Case. В результате работы этого оператора в зависимости от значения переменной⁴ выполняется один из шести операторов. Оператор Select Case используют тогда, когда в N-S-диаграмме имеется блок множественного выбора (см. выше).

6. Функция LoadPicture применяется, чтобы присвоить значение свойству Picture элемента управле-

ния Image (эта функция также может быть использована для элемента управления PictureBox). Аргумент этой функции — величина строкового типа String, которая является полным именем файла, содержащего загружаемое изображение. Если графические файлы находятся в той же папке, где форма и проект, то можно не указывать полное имя файла, а ограничиться лишь именем файла, без указания пути к нему.

7. Заметьте, что событийные процедуры, выполняемые в ответ на щелчок по каждому из двух элементов управления Image, вызывали общую процедуру по имени БросатьКости.

Вопросы и задания для проверки знаний

1. Назовите одно отличие между элементом управления Image и элементом управления PictureBox.

2. В чем различие между константой и переменной?

3. Какая инструкция объявляет переменные Кость1 и Кость2?

4. Какой оператор в Visual Basic используется для реализации множественного ветвления?

5. Какой диапазон чисел выдает функция Rnd?

6. Какие возможные целые числа можно получить из выражения $Int(Rnd * 6) + 1$?

7. С какой целью используется оператор Randomize?

8. Запишите оператор присваивания, который будет загружать картинку A:\лицо.bmp в элемент управления Image с именем imgЧеловек.

9. Разработайте программу, которая имитирует бросание монеты (воспользуйтесь изображениями “орла” и “решки”).

Литература

1. Тимофеева Н.М. Диаграммы Насси — Шнайдермана. / “В мир информатики” № 67 (“Информатика” № 2/2006).

От редакции. Пожалуйста, присылайте разработанные проекты и ответы на вопросы в редакцию. Ваша активность будет учтена при подведении итогов учебного года в нашей газете.

“Ломаем” голову

Минимальная сумма

Из шести костей домино выложить прямоугольник так, чтобы ему соответствовал пример на сложение двух чисел:

1	2	5	1
1	2	1	4
2	4	6	5

однако чтобы сумма (т.е. число в нижней строчке) была бы наименьшей.

Примечания

1. Задача должна быть решена с использованием набора неповторяющихся костей домино.

2. Кости не обязательно должны быть выложены так, как показано на рисунке.

Ответы, пожалуйста, присылайте в редакцию.

⁴ Или в зависимости от значения выражения. — Ред.

“Ломаем” голову

Кто поймал Соловья-разбойника?



Предлагаем читателям решить задачу, стоявшую перед князем Владимиром. Ответы, пожалуйста, присылайте в редакцию.

Правильные решения заданий для самостоятельной работы, предложенных в статье “Четыре приема быст-

рого счета” (она была опубликована в газете-вкладке “В мир информатики” № 75 / “Информатика” № 10/2006), прислали:

— Гайсин Рашит, г. Уфа, Республика Башкортостан, школа № 18, учитель **Искандарова А.Р.**;

— Кучина Екатерина, Мальковский Павел, Мотина Юлия и Потлова Татьяна, средняя школа села Речица Ливенского р-на Орловской обл., учитель **Потлова О.А.**

Кузнецова Евгения, г. Ярославль, школа № 33, учитель **Головина А.И.**, предложила оригинальный алгоритм возведения в квадрат трехзначных чисел, оканчивающихся на 5 (этот вопрос обсуждался в указанной статье). Опишем его на примере числа 615:

1. Рассматриваем число, образованное второй и третьей цифрой заданного числа, и делим его на 5 ($15 : 5 = 3$).

2. К количеству сотен в заданном числе приписываем цифру, полученную в пункте 1 (получаем 63).

3. Число, полученное в пункте 2, умножаем на количество сотен ($63 * 6 = 378$) — получаем первое произведение.

4. Число, образованное второй и третьей цифрами заданного числа, возводим в квадрат по правилу возведения в квадрат двухзначных чисел, оканчивающихся на 5 ($15 * 15 = 225$) — это второе произведение.

5. Записываем число, образованное первым и вторым произведением (378 225).

Уточнения приведенного алгоритма:

1) если второе произведение больше 999, то число тысяч второго произведения складываем с первым произведением и приписываем второе произведение, но уже без цифры тысяч. Например, для числа 435 имеем $35 : 5 = 7$; $47 * 4 = 188$; $35 * 35 = 1225$; $188 + 1 = 189$; результат: 189 225;

2) если в пункте 1 частное больше 10, то в пункте 2 добавляем 1 к количеству сотен в исходном числе, приписываем к сумме число единиц в частном. Особенности второго произведения — те же, что и в уточнении 1. Например, для числа 365 имеем: $65 : 5 = 13$; в пункте 2 получаем $(3 + 1) * 10 + 3 = 43$; первое произведение: $43 * 3 = 129$; второе произведение 4225; результат 133 225.

На наш взгляд, преимуществом предложенного способа является то, что в нем не приходится перемножать в уме два двухзначных числа (имеет место умножение двухзначного числа на однозначное).

⁵ Эта задача в картинках была опубликована в журнале “Квант” № 4 за 1986 г. (автор задачи — **А.П. Савин**, рисунки Э.В. Назарова).

Это полезно знать

Автоматическая память

Многие калькуляторы, даже простые, кроме регистра памяти, в который число заносится при нажатии специальной клавиши [1], снабжены памятью иного рода, которая называется *автоматической* (или *константной*). Эта память позволяет повторить последнее выполненное действие в результате нажатия клавиши $\boxed{=}$, и для пользования числом из этой памяти не нужно нажимать на какую-то специальную клавишу. Для того чтобы узнать, обеспечивает ли ваш калькулятор возможность использования автоматической памяти, выполните следующие действия:

1. Введите число 11 (например).
2. Нажмите клавишу $\boxed{=}$.
3. Несколько раз нажмите клавишу $\boxed{=}$.

Если при каждом нажатии на клавишу $\boxed{=}$ изображенное на индикаторе число будет увеличиваться на 11, то в вашем калькуляторе встроенная автоматическая память есть.

В большинстве калькуляторов константой (числом, занесенным в автоматическую память) становится: при делении — делитель, при вычитании — вычитаемое, при сложении — второе слагаемое, при умножении — второй множитель. Так, если необходимо вычесть 38 из 81, из 113, из 225 и из 1200, то после нажатия клавиш $\boxed{81} \boxed{-} \boxed{38} \boxed{=}$ значение -38 “отправляется” на хранение в константную память, после чего для вычитания 38 из других чисел, после ввода каждого из них достаточно нажимать на клавишу $\boxed{=}$.

Использовать константу можно до тех пор, пока не будет очищен индикатор калькулятора или пока в автоматическую память не будет занесено другое число.

Если после выполнения повторяющихся действий вам необходимо сделать что-то другое с числом, изображенным на индикаторе, сделать это можно легко — просто представьте, что вы только что ввели данное число.

На некоторых калькуляторах константа может не запоминаться в памяти автоматически, как это было в описанных выше случаях. На так называемых “инженерных калькуляторах” (их называют также калькуляторами для научных расчетов) для этого нужно нажимать на особую клавишу \boxed{K} . Чтобы ввести в константную память какое-либо действие и число, необходимо сначала набрать их на индикаторе, а затем нажать клавишу \boxed{K} . После этого можно пользоваться константой, как в этом примере:

$$\boxed{=} \quad \boxed{2} \quad \boxed{K}$$

$$\boxed{5} = \boxed{3}$$

$$\boxed{9} = \boxed{7}$$

Если на вашем калькуляторе клавиши \boxed{K} нет, то, для того чтобы проводить операции с константой, попробуйте нажать на клавиши операции дважды, как показано ниже:

$$\boxed{2} \quad \boxed{=} \quad \boxed{=}$$

$$\boxed{5} = \boxed{3}$$

$$\boxed{9} = \boxed{7}$$

Если не получается, ознакомьтесь с руководством по эксплуатации вашего калькулятора и узнайте, как нужно работать с константой на нем.

Задания для самостоятельной работы⁶

1. Вычислите значения:

$$1235 + 1235$$

$$12335 + 1235 + 1235$$

$$1235 + 1235 + 1235 + 1235$$

$$1235 + 1235 + 1235 + 1235 + 1235$$

Примечание. Операцию умножения не использовать.

2. Вычислить первый, второй, третий, ..., десятый члены прогрессий:

1) арифметической, у которой первый член и разность соответственно равны 10,7 и 2,7;

2) геометрической, у которой первый член и знаменатель соответственно равны 1,8 и 2,1;

3) арифметической, у которой первый член и разность соответственно равны 128,5 и $-12,1$;

4) геометрической, у которой первый член и знаменатель соответственно равны 431,8 и $1/3$.

3. Вычислить значения 2^{10} и 2^{20} .

4. Когда растение посадили в горшок, то его длина от поверхности земли в горшке составляла 1 см. Какова будет длина растения через 7 дней, если за 1 день она удваивается?

5. “Задача о шахматной доске”.

Существует легенда, согласно которой изобретатель шахмат (шахматы впервые появились в Индии в V веке) в ответ на предложение раджи выбрать награду за свое изобретение сказал, что он хотел бы получить столько зерен риса, сколько уместится на шахматной доске, если на первую ее клетку положить 1 зерно, на вторую — 2 зерна, на третью клетку — 4 зерна, на

⁶ Все задания выполнить на калькуляторе, используя автоматическую память.

четвертую — 8 и так далее, удваивая число зерен на каждой следующей клетке. Раджа обрадовался, что награждение обойдется ему так дешево, и согласился. Смог ли раджа выполнить свое обещание?

В заключение — полезный совет по использованию константной памяти. Если вам необходимо пересчитать-

вать много различных предметов, то для этого удобно ввести в калькулятор +1 (Нажать $\boxed{+}$ и $\boxed{1}$) и с каждым предметом нажимать клавишу $\boxed{=}$.

Литература

1. Память калькулятора. / "В мир информатики" № 62 ("Информатика" № 21/2005).

Games.exe

Компьютерные фокусы на отгадывание чисел

Продемонстрируйте своим товарищам несколько компьютерных фокусов. Для этого подготовьте ряд несложных компьютерных программ.

Фокус № 1

Предложите кому-нибудь задумать число, прибавить к нему 3, умножить полученную сумму на 6^1 , вычесть из этого произведения задуманное число, затем вычесть еще 8 и, наконец, разделить все это на 5.

Соответствующая программа на школьном алгоритмическом языке имеет вид:

```

алг Фокус_1
нач цел x, n
  вывод нс, "Задумайте целое число, кото-
    рое я отгадаю"
  | Приостановка программы до нажатия
    любой клавиши
нц
  |
  кц при inkey() <> ""
  вывод нс, "Прибавьте к нему 3"
  | В дальнейшем фрагмент программы,
    относящийся к ее приостановке,
  | будет представлен в виде многоточия (...)
  ...
  вывод нс, "Полученную сумму умножьте
    на 6"
  ...
  вывод нс, "Из полученного произведения
    вычтите задуманное число"
  ...
  вывод нс, "Из результата вычтите число
    8"
  ...
  вывод нс, "И, наконец, разделите все это
    на 5"
  ...
  вывод нс, "Сколько получилось?"
  ...
  вывод нс, "Введем полученный результат"
ввод n

```

¹ Можно пользоваться калькулятором.

```

x := n - 2
Очистка_экрана
вывод нс, "Вы задумали число", x
...

```

КОН

— где Очистка_экрана — стандартная процедура очистки экрана.

Этот фокус можно продемонстрировать и без использования компьютера. После того как ваш товарищ назовет полученный результат N , вы можете назвать задуманное им число (оно равно $N - 2$)².

Конечно, после нескольких демонстраций такого фокуса вас могут "раскусить" (почти "расфокусить" ©) — слишком уж очевиден секрет фокуса. Труднее это сделать при демонстрации следующего фокуса.

Фокус № 2

```

алг Фокус_2
нач цел x, y, n
  вывод нс, "Задумайте два целых числа "
  вывод "(каждое не более 99), "
  вывод "которые я отгадаю"
  ... | см. выше
  вывод нс, "Первое из них умножьте на 2"
  ...
  вывод нс, "К полученному произведению
    прибавьте 5"
  ...
  вывод нс, "Результат умножьте на 50"
  ...
  вывод нс, "К произведению прибавьте
    второе число"
  ...
  вывод нс, "И, наконец, вычтите из
    результата "
  вывод нс, "количество дней в году
    (невисокосном)"
  вывод нс, "Сколько получилось?"
  ...
  вывод нс, "Введем полученный результат"
ввод n
  x := div(n + 115, 100)
  y := div(n + 115, 100)
  вывод нс, "Вы задумали числа ", x, "
    и ", y

```

КОН

² Докажите это самостоятельно, обозначив задуманное число x и проведя соответствующие преобразования. Доказательство пришлите в редакцию.

Следующие фокусы связаны с отгадыванием не исходного числа, а результата расчетов. Для каждого из фокусов приведем перечень сообщений, выводимых на экран монитора при их демонстрации.

Фокус № 3

1. Задумайте число.
2. Умножьте на 2.
3. Прибавьте 48.
4. Разделите на 2.
5. Вычтите задуманное число.
6. У вас получилось 18.

Фокус № 4

1. Задумайте однозначное число.
2. Умножьте на 3.
3. Прибавьте 3.
4. Умножьте на 3.
5. Найдите сумму цифр полученного числа.
6. У вас получилось 9.

Фокус № 5

1. Задумайте трехзначное число.
2. Припишите к нему справа то же самое число.
3. Полученное шестизначное число разделите на 11.
4. Результат разделите на задуманное число.
5. У вас получилось 91.

Фокус № 6

1. Задумайте число.
2. Умножьте на 6.
3. Прибавьте 50.
4. Разделите на 2.
5. Вычтите 4.
6. Разделите на 3.
7. Вычтите задуманное число.
8. У вас получилось 7.

Фокус № 7

1. Задумайте число.
2. Умножьте на 2.
3. Прибавьте 12.
4. Разделите на 2.
5. Вычтите 5.
6. Вычтите задуманное число.
7. У вас получилось 1.

Фокус № 8

1. Задумайте число.
2. Умножьте на 6.
3. Прибавьте 22.
4. Разделите на 2.
5. Вычтите 2.
6. Разделите на 3.
7. Вычтите задуманное число.
8. У вас получилось 3.

Фокус № 9

1. Задумайте число.
2. Умножьте на 2.
3. Прибавьте 14.
4. Разделите на 2.
5. Вычтите 4.
6. Вычтите задуманное число.
7. У вас получилось 3.

Фокус № 10

1. Задумайте трехзначное число.
2. Припишите к нему справа то же самое число.
3. Полученное шестизначное число разделите на 13.
4. Результат разделите на задуманное число.
5. У вас получилось 77.

Фокус № 11

1. Задумайте число.
2. Умножьте на 2.
3. Прибавьте 36.
4. Разделите на 2.
5. Вычтите 8.
6. Вычтите задуманное число.
7. У вас получилось 10.

Фокус № 12

1. Задумайте число.
2. Умножьте на 6.
3. Прибавьте 46.
4. Разделите на 2.
5. Вычтите 5.
6. Вычтите задуманное число.
7. У вас получилось 6.

Программы для демонстрации фокусов составьте самостоятельно. Учитывая, что в них будут повторяться одни и те же фразы ("Задумайте число", "Задумайте трехзначное число", "Умножьте на...", "У вас получилось..." и т.п.), целесообразно:

1) описать эти фразы как глобальные константы [1]:

```
лит Задум, Задум3, Умн, Разд, Раздб, Приб,
      Выч, ВычЗадум, Припис, НайдСум, Получ
Задум := "Задумайте число"
Задум3 := "Задумайте трехзначное число"
Умн := "Умножьте на "
Разд := "Разделите на "
Раздб := "Полученное шестизначное число
разделите на "
Приб := "Прибавьте "
Выч := "Вычтите "
ВычЗадум := "Вычтите задуманное число"
Припис := "Припишите к нему справа то же
самое число"
НайдСум := "Найдите сумму цифр полученного
числа"
Получ := "У вас получилось "
```

2) каждый фокус оформить в виде отдельной процедуры. Например, для фокуса № 3 соответствующая процедура имеет вид:

```
алг Фокус_3
```

```
нач
```

```
ВЫВОД нс, Задум
... | см. выше
ВЫВОД нс, Умн, 2
...
ВЫВОД нс, Приб, 48
...
ВЫВОД нс, Разд, 2
...
ВЫВОД нс, Выч, 6
```

```
...
вывод нс, ВычЗадум
...
вывод нс, Получ, 18
...
```

кон

3) в главной части программы предусмотреть случайный выбор того или иного фокуса, а затем вызвать соответствующую процедуру, используя оператор варианта (выбора):

Глобальные константы литерного (строкового) типа

```
лит Задум, Задум3, Умн, Разд, Раздб,
    Приб, Выч, ВычЗадум, Припис, Получ
Задум := "Задумайте число"
```

...

```
алг Фокусы | Главная часть программы
```

```
нач цел нф | номер фокуса
```

```
вывод нс, "Выберем фокус"
| Приостановка программы до нажатия любой
| клавиши
```

нц

|

```
кц_при inkey() <> ""
```

```
нф := 3 + rnd(10) | Случайное число в
| интервале от 3 до 12
```

выбор

```
при нф = 3: Фокус_3
```

```
при нф = 4: Фокус_4
```

...

```
при нф = 12: Фокус_12
```

все

```
| Приостановка программы до нажатия любой
| клавиши
```

нц

|

```
кц_при inkey() <> ""
```

кон

```
алг Фокус_3
```

нач

...

Разработанные программы на известном вам языке программирования присылайте в редакцию. Фамилии всех приславших будут опубликованы, а лучшие ответы мы поощрим.

Литература

1. Школа программирования: Тематический выпуск газеты-вкладки "В мир информатики" / "Информатика" № 11/2005.



«Портфолио»

Фестиваль исследовательских
и творческих работ учащихся



Издательский дом «Первое сентября» объявляет о проведении в 2006/07 учебном году Второго всероссийского фестиваля «Портфолио» и приглашает принять в нем участие учащихся учреждений начального, среднего и дополнительного образования и их педагогов.

Участвуя в фестивале, учащиеся могут формировать общедоступное портфолио своих работ. Также формируется портфолио педагога, в которое входят работы учащихся, выполненные под его руководством.

Все материалы будут опубликованы. По результатам фестиваля будут изданы: книга — сборник тезисов (описаний) работ и компакт-диски с полными версиями работ. Полные версии работ также публикуются на сайте фестиваля <http://portfolio.1september.ru>, который является одним из разделов сайта Издательского дома «Первое сентября» — самого популярного образовательного ресурса русскоязычного Интернета. **Книги и компакт-диски будут высланы всем участникам. Все ученики и их руководители будут отмечены дипломами.**

Заявки на участие принимаются с 1 июля по 15 декабря 2006 г.

Заявки можно подавать на сайте <http://portfolio.1september.ru> или по почте, используя бланки, публикуемые в газетах.

РАЗДЕЛЫ ФЕСТИВАЛЯ

- Научно-исследовательские работы:
 - Астрономия, космонавтика и авиация
 - Биология
 - География
 - История, археология
 - Искусствоведение
 - Лингвистика
 - Литературоведение
 - Математика
 - Религиоведение
 - Экономика, социология и право
 - Физика
 - Химия
 - Здоровье человека, психология
 - Физкультура и спорт
 - Экология
- Художественное творчество
- Техническое творчество
- Информационные технологии
- Литературное творчество
- Музыкальное творчество
- Краеведение
- В помощь учителю (дополнительный раздел)

С

«СОЛОН»

РАДИОТЕХНИЧЕСКАЯ
КОМПЬЮТЕРНАЯ
ЛИТЕРАТУРА

Профильное обучение информатике

С 2003 года в школах России проводится эксперимент по внедрению профильного обучения учащихся.

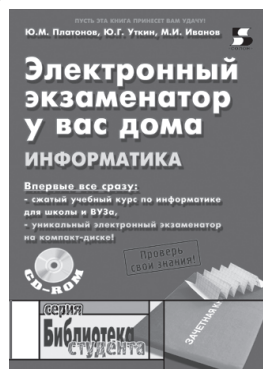
Издательство «СОЛОН-ПРЕСС» имеет богатейший опыт

издания профессиональной компьютерной и технической литературы. Мы шагаем в ногу со временем, отслеживая все новейшие разработки и технологии в данных областях, привлекая к сотрудничеству лучших отечественных и зарубежных специалистов. Кроме того, наше издательство начало выпуск литературы по информатике в помощь школьному учителю.



Комплект «Рабочих тетрадей по информатике» авторов М. Бардо и А. Лишина можно использовать как дополнение к учебным пособиям других авторов, а также как самостоятельные пособия. Отдельно готовятся к выпуску методические рекомендации для учителей.

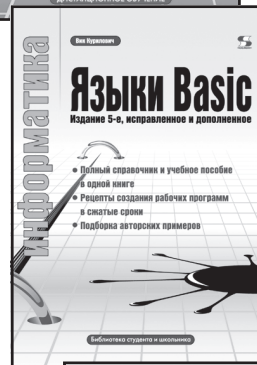
Для базового курса информатики в качестве дополнительного учебного пособия можно порекомендовать книгу Ю. Платонова «Электронный экзаменатор у вас дома. Информатика».



Материал книги подобран так, чтобы систематизировать разрозненные знания, полученные из различных учебников, а прилагаемая на компакт-диске оригинальная программа является своеобразным «электронным экзаменатором» для проверки знаний по информатике.



По книгам «Практикум Web-дизайна» Т. Третьяк, М. Кубаревой, «Курс Delphi для начинающих. Полигон нестандартных задач» А. Ремнева, С. Федотовой учатся школьники Северо-Западного округа Москвы.



Книга В. Куриловича «Информатика. Языки Basic» фактически является бестселлером, выдержала не одно переиздание, принимала участие в ярмарке педагогических достижений, где стала одним из лауреатов в номинации «Учебные пособия и методические разработки».



Отличительной особенностью наших книг – является преемственность – с одинаковым успехом они могут изучаться и старшеклассниками, и учащимися колледжей, и слушателями подготовительных курсов и студентами вузов, и даже инженерно-техническими работниками. Они способны сократить существующую пропасть между школой и вузом, что входит в цели и задачи профильного обучения.



АЛЬЯНС-КНИГА КТК
ТОРГОВО-ИЗДАТЕЛЬСКИЙ ХОЛДИНГ

По вопросам приобретения книг обращайтесь по тел.: (495) 258-91-94, 258-91-95, 258-91-96
сайт: www.abook.ru, e-mail: books@alians-kniga.ru

“КАК ЭТО ДЕЛАЮ Я”

Методический конкурс для учителей информатики

В прошлом номере мы опубликовали задание 8-го тура “Программы, которые мы выбираем. Часть I: организация учительского труда”. В нем шла речь о программах, которые учителя информатики используют для облегчения и автоматизации повседневного учительского труда. Сегодня мы хотим предложить вам поделиться опытом использования программного обеспечения в учебном процессе.

ЗАДАНИЕ ДЕВЯТОГО ТУРА “ПРОГРАММЫ, КОТОРЫЕ МЫ ВЫБИРАЕМ”. Часть II: программное обеспечение учебного процесса

В этом туре нашего конкурса, без сомнения, может принять участие каждый учитель информатики. Уж без чего другого, а без программ нам — никуда! Языки программирования, редакторы, оболочки, операционные системы, наконец, — мы в этом живем и этому учим.

Какое программное обеспечение используете вы? Почему вы выбрали именно то, что выбрали? Что посоветуете коллегам?

В данном случае нас интересует равным счетом все — и самый обычный Turbo Pascal, и стандартный офисный набор, и оригинальные тестирующие системы (см., например, статью В.А. Матюхина в № 18), и оболочки, и почтовые серверы для локальной сети, посредством которых многие наши коллеги организуют электронную почту в школе.

Пожалуйста, не думайте, что если вы не используете “ничего особенного”, то и писать об этом не надо. Ведь общая картина использования ПО в школах очень интересна. На нее будут ориентироваться и разработчики, и поставщики программ.

Ждем ваших писем!

ФОРМАТ ПРЕДСТАВЛЕНИЯ КОНКУРСНЫХ МАТЕРИАЛОВ

Материалы принимаются только в электронном виде в формате Microsoft Word. Объем основного текста (с учетом пробелов) — не более 6000 знаков. Дополнительные материалы можно оформлять в виде приложений. Объем приложений не ограничен. При описании нестандартного (малоизвестного) программного обеспечения, пожалуйста, указывайте ссылки и источники его получения.

Материалы можно присылать на электронных носителях или по электронной почте.

Почтовый адрес: 121165, Москва, ул. Киевская, д. 24, “Первое сентября”, “Информатика”.

Электронный адрес: inf@1september.ru.

В теме (subject) электронных писем, пожалуйста, указывайте: “Методический конкурс, девятый тур”.

Вместе с материалами, пожалуйста, присылайте краткую информацию о себе, в обязательном порядке включающую: фамилию, имя, отчество, полное название учебного заведения, в котором вы преподаете, стаж работы. Пожалуйста, указывайте также максимально полную контактную информацию: полный почтовый адрес (с индексом), электронный адрес, телефоны (с кодом населенного пункта). (Эта информация не будет опубликована, но она может потребоваться редакции для оперативной связи.)

КЛЮЧЕВЫЕ СРОКИ

Срок отправки материалов девятого тура — до 30 ноября 2006 г. (для “бумажных” писем дата отправки фиксируется по штемпелю предприятия-отправителя).

Результаты девятого тура будут опубликованы в № 4/2007. На сайте “Информатики” материалы будут размещаться по мере поступления.

Задание следующего тура будет опубликовано в № 21.

ИЗДАТЕЛЬСКИЙ ДОМ «ПЕРВОЕ СЕНТЯБРЯ» главный редактор — А.С. Соловейчик

ГАЗЕТЫ
ИЗДАТЕЛЬСКОГО ДОМА
Первое сентября
гл. ред. — Е.В. Бирюкова,
индекс подписки — 32024;

Английский язык
гл. ред. — Е.В. Громушкина,
индекс подписки — 32025;

Библиотека в школе
гл. ред. — О.К. Громова,
индекс подписки — 33376;

Биология
гл. ред. — Н.Г. Иванова,
индекс подписки — 32026;

География
гл. ред. — О.Н. Коротова,
индекс подписки — 32027;

Дошкольное образование
гл. ред. — М.С. Аромштам,
индекс подписки — 33373;

Здоровье детей
гл. ред. — Н.В. Семина,
индекс подписки — 32033;

Информатика
гл. ред. — С.Л. Островский,
индекс подписки — 32291;

Искусство
гл. ред. — М.Н. Сартан,
индекс подписки — 32584;

История
гл. ред. — А.Л. Савельев,
индекс подписки — 32028;

Литература
отв. сек. — С.Ф. Дмитренко,
индекс подписки — 32029;

Математика
и. о. гл. ред. — Л.О. Рослова,
индекс подписки — 32030;

Начальная школа
гл. ред. — М.В. Соловейчик,
индекс подписки — 32031;

Немецкий язык
гл. ред. — М.Д. Бузова,
индекс подписки — 32292;

Русский язык
гл. ред. — Л.А. Гончар,
индекс подписки — 32383;

Спорт в школе
гл. ред. — О.М. Леонтьева,
индекс подписки — 32384;

Управление школой
гл. ред. — Я.А. Сартан,
индекс подписки — 32652;

Физика
гл. ред. — Н.Д. Козлова,
индекс подписки — 32032;

Французский язык
гл. ред. — Г.А. Чесновицкая,
индекс подписки — 33371;

Химия
гл. ред. — О.Г. Блохина,
индекс подписки — 32034;

Школьный психолог
гл. ред. — И.В. Вачков,
индекс подписки — 32898.

Гл. редактор
С.Л. Островский
Зам. гл. редактора
А.И. Сенокосов
Редакция
Е.В. Андреева
Д.М. Златопольский (редактор
вкладки “В мир информатики”)
Л.Н. Картвелишвили
С.Б. Кишкина
Н.П. Медведева
Ю.А. Первин (редактор вкладки
“Началка”)
Корректор Дизайн и верстка
Е.Л. Володина Н.И. Пронская

©ИНФОРМАТИКА 2006
Выходит два раза в месяц
При перепечатке ссылка
на ИНФОРМАТИКУ обязательна,
рукописи не возвращаются

Адрес редакции и издателя:

Киевская, 24, Москва, 121165
тел. 8-499-249-48-96
Отдел рекламы: 8-499-249-98-70

ИНДЕКС ПОДПИСКИ

для индивидуальных подписчиков 32291
комплекта изданий 32744

Тел.: 8-499-249-31-38, 249-33-86. Факс 8-499-249-31-84

Учредитель: ООО “Чистые пруды”

Зарегистрировано в Министерстве РФ по делам печати. ИП
№ 77-7230 от 12.04.2001.
Отпечатано в ОИД “Медиа-Пресса”,
ул. Правды, 24, Москва, ГСП-3, А-40, 125993
Тираж 6500 экз.
Срок подписания в печать по графику 15.09.2006.
Номер подписан 15.09.2006.
Заказ № 615520
Цена свободная

Internet: inf@1september.ru
WWW: <http://www.1september.ru>