

Суп будущего?

В октябре в русскоязычном Интернете и в среде IT-специалистов активно обсуждалась покупка новой интернет-компанией “Суп” русского сегмента популярного блог-сервиса Livejournal (в просторечии — ЖЖ). Окончательные параметры этой сделки широкой публике не до конца ясны и сейчас (прежде всего, по-видимому, потому, что не до конца ясны они и покупателю — “Супу”, и продавцу — американской компании Six Apart). Пресс-конференция создателя ЖЖ Брэда Фицпатрика, который специально для этого приезжал в Москву, мягко говоря, ничего не прояснила. Все это можно было бы принять к сведению или даже вовсе проигнорировать (в рамках “Информатики”), если бы не одно обстоятельство — вопрос стремительного и реального формирования информационного (со)общества имеет и к нам, преподавателям информатики, и к нашим ученикам самое непосредственное отношение.

ЖЖ и связанные с ним события — не первый и не единственный звоночек. В ряду значимых проектов подобного рода нельзя пройти мимо Википедии. Если ЖЖ можно условно отнести к категории “трепа” (хотя это, конечно, неправильно!), то Википедия, со всеми ее неоднозначностями, — продукт нашего образовательного поля.

В одной из осенних колонок я писал о Геннадии Викторовиче Лебедеве. У него была определенная позиция, чему можно учить, а чему учить не следует. Одним из важнейших критериев при этом было наличие задач. Если есть содержательные задачи — этому можно (не значит — нужно) учить. Если задач нет — не надо “лить воду”. Это, конечно, немного “экстремистская” точка зрения, но она вполне согласуется с моим личным педагогическим опытом. Не имея достаточного количества задач, учить очень сложно. Буду очень благодарен, если кто-то посоветует мне, где взять содержательные задачи, например, по основам социальной информатики, пока же я нашел для себя следующую формулу: такого рода вещам я не учу, я им “образовываю”. Такое образование носит не столько предметный, сколько общекультурный характер. Поскольку мне трудно себе представить, что вопросы по социальной информатике когда-нибудь войдут, например, в ЕГЭ, выбранная формула меня устраивает. (А если войдут, я буду очень рад — увижу, наконец, задачи!)

Безусловно понимая и принимая позицию Г.В. Лебедева, я тем не менее считаю, что реальные проявления информационного общества должны стать обязательным элементом информатического образования наших учеников. (На мой-то личный взгляд, эти вопросы куда важнее основ социальной информатики — ну, да, со стандартом не спорят.) Тем более что, если не задачи, то образовательные проекты в этой области известны и активно развиваются (приведу лишь один пример — основанный на вики-технологиях проект “Летописи” www.letopisi.ru). Мы, учителя информатики, не имеем права игнорировать реальность, имеющую к нам непосредственное отношение. Если не мы, то кто?

С.Л. Островский, главный редактор

СОДЕРЖАНИЕ НОМЕРА

ИНФОРМАЦИЯ

Набор слушателей на очно-заочные курсы повышения квалификации (для жителей Москвы и Московской области).

Набор слушателей на дистанционные курсы повышения квалификации. Второй поток 2

Фестиваль педагогических идей “Открытый урок” 47

ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ

В.Ф. Бурмакина, И.Н. Фалина. Как готовиться к тестированию по проверке ИКТ-компетенции школьников.

Лекция 7. Контрольная работа № 2 3–10

ПРОФИЛЬНАЯ ШКОЛА. ЭКЗАМЕНЫ

Е.А. Еремин, А.П. Шестаков. Примерные ответы на профильные билеты 11–15

ТЕЛЕКОММУНИКАЦИИ

А.А. Дуванов. Азы информатики. Выходим в Интернет 16–21

ОЛИМПИАДЫ

А.А. Ильенко. Брянская областная командная олимпиада среди школьников по информационным и коммуникационным технологиям 22–26

I Всероссийская заочная олимпиада школьников по информатике 2006/2007 учебного года 27

ВНЕКЛАССНАЯ РАБОТА

О.В. Королева. Игра “Сто к одному” 28

“НАЧАЛКА” № 16

Газета-клуб для всех, кто учит информатике маленьких детей

Н.Г. Иванова, М.А. Плаксин, О.Л. Русакова. Материалы конкурса “ТРИЗформашка-2006”. Часть III 29–36

“В МИР ИНФОРМАТИКИ” № 82

Газета для пытливых учеников и их талантливых учителей

Эксперименты

Е.А. Еремин. Еще раз о программе Debug .. 37–40

Моделирование

Л.Н. Медведев. Модель колеса. Часть 2 40–42

Задачник

Ответы, решения, разъяснения 43–46

Конкурс

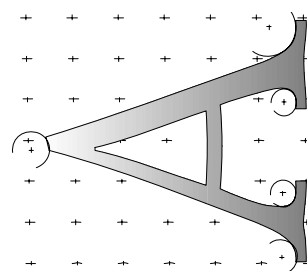
Конкурс № 53 для учащихся 46

“КАК ЭТО ДЕЛАЮ Я”

Методический конкурс для учителей информатики ... 48

№ 23 (529)

1–15 декабря 2006



Методическая газета для учителей информатики

ИНФОРМАТИКА



ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ «ПЕРВОЕ СЕНТЯБРЯ»
 ГАЗЕТА «ИНФОРМАТИКА»
 ОТДЕЛЕНИЕ ПЕДАГОГИЧЕСКОГО ОБРАЗОВАНИЯ ФГП МГУ ИМ. М.В. ЛОМОНОСОВА

ОБЪЯВЛЯЮТ НОВЫЙ НАБОР СЛУШАТЕЛЕЙ НА ТЕКУЩИЙ УЧЕБНЫЙ ГОД

НОВОЕ ПРЕДЛОЖЕНИЕ!

для жителей Москвы и Московской области

ОЧНО-ЗАОЧНЫЕ КУРСЫ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

Объявляется набор на очно-заочные курсы повышения квалификации с нормативным сроком освоения учебного материала 72 часа (36 часов очно, 36 — заочно). Занятия будут проходить в феврале–апреле 2007 г., в удобное время во второй половине дня, 1 раз в неделю. Наш комфортный учебный центр расположен рядом со ст. м. «Киевская».

Мы предлагаем два курса для учителей информатики:

| Код | Курс |
|---------|--|
| 107-001 | А.А. Дуванов, С.Л. Островский. Основы web-дизайна и школьного сайтостроительства |
| 107-002 | Ю.А. Первин. Обоснования и методика школьного курса информатики |

Мы также предлагаем два общепедагогических курса, предназначенных для всех работников образования:

| | |
|---------|---|
| 121-001 | В.М. Букатов. Режиссура урока в современной школе |
| 121-002 | А.П. Ершова. Театральное мастерство в работе современного учителя |

Стоимость обучения — 3900 руб. за один курс. Для тех, кто ранее обучался (или в настоящее время обучается) на наших дистанционных курсах, — 3400 руб.

Заявки на очно-заочные курсы можно подавать по телефону: (499) 249-47-82, (495) 249-52-53 и на сайте <http://edu.1september.ru>

Регистрация слушателей производится с 1 ноября 2006. Количество мест в группах ограничено.

ВТОРОЙ ПОТОК

для всех работников образования вне зависимости от места их проживания

ДИСТАНЦИОННЫЕ КУРСЫ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

Объявляется набор на **второй поток 2006/2007 учебного года**. Курсы проводятся в режиме дистанционного обучения. Взаимодействие со слушателями производится посредством обычной или, при наличии у слушателя возможности, электронной почты. Нормативный срок освоения учебного материала — 72 часа. Начало занятий — январь 2007 г.

После успешного окончания курсов слушатели получают удостоверение установленного образца о прохождении курсов повышения квалификации от Педагогического университета «Первое сентября» и Отделения педагогического образования ФГП МГУ им. М.В. Ломоносова.

Стоимость обучения составляет 990 рублей за один курс.

На втором потоке 2006/2007 учебного года мы предлагаем три дистанционных курса для учителей информатики:

| Код | Курс |
|--------|---|
| 07-001 | И.Г. Семакин. Информационные системы в базовом и профильном курсах информатики |
| 07-002 | Е.В. Андреева. Методика обучения основам программирования на уроках информатики |
| 07-006 | А.А. Дуванов. Основы web-дизайна и школьного сайтостроительства |

Мы также предлагаем общепедагогический курс, предназначенный для всех работников образования:

| | |
|--------|--|
| 21-001 | С.С. Степанов. Теория и практика педагогического общения |
|--------|--|

Для зачисления на курсы необходимо прислать в Педагогический университет «Первое сентября» заявку. Пожалуйста, используйте только приведенный ниже бланк или его ксерокопию. Регистрация слушателей производится с 1 ноября по 31 декабря 2006 г. (дата отправки заявки фиксируется по почтовому штемпелю предприятия-отправителя). Вам будет выслан комплект документов с подробной информацией о курсах и счетом для оплаты. Вы оплатите счет, если вас устроят предлагаемые условия (факт подачи заявки ни к чему не обязывает).

Заявки следует направлять по адресу: Педагогический университет «Первое сентября», ул. Киевская, д. 24, Москва, 121165.

Заявки также можно подавать on-line на сайте <http://edu.1september.ru> Справки по тел.: (499) 249-47-82, e-mail: edu@1september.ru

Прошу выслать мне комплект документов для зачисления на ДИСТАНЦИОННЫЕ КУРСЫ повышения квалификации

ФАМИЛИЯ

ИМЯ

ОТЧЕСТВО

ИНДЕКС

АДРЕС

Телефон (с кодом города): () _____ E-mail (если есть): _____

Место работы: _____ Должность: _____

ВНИМАНИЕ! К обучению на курсах повышения квалификации допускаются сотрудники образовательных учреждений, работающие по соответствующей специальности.

Если вы обучались в прошлом учебном году на наших курсах, укажите ваш идентификатор:

07-23

Я хочу пройти обучение по дистанционным курсам (укажите коды выбранных вами курсов):

 —
 —
 —

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-компетенции 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 информацией". 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 7.

Описание познавательных деятельности, составляющих компетентность "создание информации". Примеры заданий на формирование этой компетентности

Когнитивная деятельность "создание информации" включает в себя практически все виды деятельности, которые мы уже рассматривали в рамках ИКТ-компетентности. В соответствии с определением ИКТ-компетентности эта познавательная деятельность состоит из следующих умений:

Характеристика познавательного действия "создание информации"

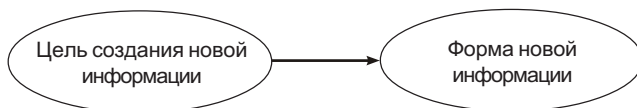
Как правило, под деятельностью "создание информации" понимается написание каких-либо рефератов, докладов и т.д., и, казалось бы, перечислен-

Таблица 1

| | |
|----------|--|
| Создание | Умение вырабатывать рекомендации по решению конкретной проблемы на основании полученной информации, в том числе противоречивой |
| | Умение сделать вывод о нацеленности имеющейся информации на решение конкретной проблемы |
| | Умение обосновать свои выводы |
| | Умение сбалансированно осветить вопрос при наличии противоречивой информации |
| | Структурирование созданной информации с целью повышения убедительности выводов |

ные в табл. 1 умения выходят за рамки описываемой деятельности. Однако это не так. В данной деятельности акцент делается не на *форму* представления новой (созданной) информации, а на *цель*, ради которой создается новая информация. Естественно, что любая вновь созданная информация должна быть представлена в какой-то форме. Более того, цель создания новой информации определяет форму ее представления. Выбор формы представления новой информации в большей степени относится к умению “сообщение или передача информации”. Это отдельное, важное умение, которое будет рассмотрено в лекции 8.

Деятельность — создание информации



Как правило, цель создания информации к началу выполнения данной деятельности задана извне или сформулирована самим человеком. Так как мы занимаемся “препарированием” ИКТ-компетентности, вычлениением элементарных операций в описываемых деятельности, то будем считать, что любой человек, учащийся в том числе, выполняющий деятельность по созданию новой информации, работает с ранее отобранными источниками информации. А вот *как* он работает с уже отобранными источниками в соответствии с поставленной целью, и составляет сущность этого умения. Очевидно, что цель создания новой информации определяет, или уточняет, специфику этой деятельности — именно поэтому определение деятельности “создание информации” разделено на подпункты.

Описание когнитивных действий, составляющих компетентность “создание информации”

Мы уже неоднократно писали, что ИКТ-компетентность — это общеучебное умение работать с информацией, представленной в электронном виде. Соответственно, формирование этого умения должно проходить на всех школьных уроках, а не только на уроках информатики. Это замечание особенно актуально при рассмотрении формирования умения по созданию информации. К сожалению, формированию умения создания новой информации в наших школах уделяется очень мало внимания, хотя доля рефератов, докладов, принимаемых учителями в электронном (или отпечатанном) виде, год от года растет. В подтверждение такого пессимистического вывода можно привести следующие факты (которые конечно же являются лишь косвенным подтверждением высказанного утверждения):

1) по оценкам преподавателей вузов, студенты младших курсов не умеют писать рефераты в соответствии с предъявляемыми (к рефератам) требованиями; практически во всех вузах страны на младших курсах введен специальный курс, в рамках которого студентов учат навыкам поиска информации в библиотечных каталогах и картотеках, знакомят с требованиями оформления рефератов, курсовых, статей и т.д.;

2) в исследовании, выполненном Центром социологии образования РАО [1], приведена таблица использования обучающих и демонстрационных компьютерных программ учителями, преподающими различные предметы (%)

| | |
|---------------------|------|
| Информатика | 79,2 |
| Математика | 8,5 |
| Физика | 20,5 |
| Химия | 28,9 |
| Биология | 26,3 |
| География | 7,1 |
| Общественные науки | 10,4 |
| Иностранный язык | 12,7 |
| Русский, литература | 12,1 |

Анализ этой таблицы наталкивает на вывод: если учитель не использует в своей педагогической практике ИКТ-технологии, то он и не уделяет должного внимания формированию ИКТ-компетентности своих учащихся. Поэтому нам хотелось, чтобы вы, уважаемые учителя информатики, по возможности обратили внимание руководства своей школы на существующую проблему (проблему формирования ИКТ-компетентности) и поделились со своими коллегами — учителями-предметниками — материалами наших лекций: быть может, эти лекции и приводимые задания их заинтересуют и заставят задуматься о проблеме формирования ИКТ-компетентности.

Все нижеприведенные учебные задания, направленные на формирование умения создания информации, имеют бытовой контекст. На сегодняшний момент мы затрудняемся привести полностью разработанные задания с информатическим контекстом. Для выработки этого умения наиболее целесообразно использовать активные методы обучения: деловые и ролевые игры, метод case-study, мозговой штурм и т.д. Для любого учебного задания необходимо подготовить несколько информационных источников, четко сформулировать цель задания, определить форму сдаваемого задания, т.е. форму информации, которая будет создана.

Умение вырабатывать рекомендации по решению конкретной проблемы на основании полученной информации, в том числе противоречивой

Пример 1. Задание “Экскурсия” рекомендуется выполнить с использованием метода case-study. Метод case-study — один из самых старых и испытанных активных методов обучения принятия решений. Для проведения этого занятия класс можно разбить на группы “ищущих решение” и “экспертов”. Каждой группе даются заранее подготовленные одинаковые информационные источники. Группа “ищущих решение” должна подготовить рекомендации по решению проблемы, группа “экспертов” — оценить это решение.

Проводить игру можно в нескольких вариантах:

- заранее сформировать группы-пары (“ищущие решение” и “эксперты”). По окончании определенного отрезка времени группа-пара начинает обсуждать представленные рекомендации;
- каждая группа “ищущих решение” выполняет задание в своем темпе, и только после завершения выполнения задания начинается обсуждение со свободной группой “экспертов”.

Задание. Ваш класс в ближайшие выходные едет на экскурсию в город Чудесное на 2 дня. Вам поручено составить расписание на первый день экскурсии. Вы располагаете следующей информацией:

1) в город Чудесное вы можете добраться только на автобусе, но при этом вам придется сначала доехать до станции Отдых на электричке, а затем уже на автобусе;

2) расписание движения электричек от вашего города Славное до станции Отдых;

3) расписание движения автобусов от станции Отдых до г. Чудесное;

4) статья местной газеты с описанием производимых работ на станции Отдых;

5) регистрация в гостинице “Маяк”, в которой вы должны поселиться, начинается с 12 часов дня;

6) обзорная автобусная экскурсия по г. Чудесное назначена на 17.00;

7) прогноз погоды на ближайшие выходные: сильный ветер, возможен мокрый снег.

Проанализировав имеющуюся информацию, вы должны составить комфортное расписание первого экскурсионного дня. Все в ваших руках: время отъезда, время в пути и т.д.

Информационные источники:

1. Расписание движения электричек от г. Славное до ст. Отдых
2. Расписание движения автобусов от ст. Отдых до г. Чудесное

| Славное — Сергово | | |
|-------------------------|-----------------------|---|
| Отправление из Славного | Прибытие на ст. Отдых | Примечание |
| 5-10 | 7-06 | Ежедневно |
| 6-52 | 8-44 | Ежедневно |
| 7-44 | 9-37 | Ежедневно до Сергово |
| 8-13 | 10-09 | Ежедневно |
| 9-43 | 11-40 | Ежедневно |
| 10-45 | 12-46 | По рабочим дням до Путевого, по выходным до Сергово |
| 12-09 | 14-07 | По выходным и праздничным дням |
| 13-30 | 15-28 | Ежедневно |
| 14-12 | 16-08 | По пятницам |
| 14-29 | 16-28 | Ежедневно до Сергово |
| 15-20 | 17-16 | Ежедневно |
| 17-37 | 19-36 | Ежедневно до Сергово |
| 18-31 | 20-27 | Ежедневно |
| 19-19 | 21-10 | По пятницам |
| 19-30 | 21-28 | Ежедневно |
| 20-20 | 22-16 | Ежедневно |
| 21-33 | 23-32 | По пятницам и воскресеньям |

Отдых — Чудесное: 7-45, 9-35, 13-10, 16-05, 18-05 (время в пути — 35 минут).

3. *Статья из газеты “Родные просторы” от 14 октября 2006 года.*

Этой осенью из сельского бюджета были выделены деньги на реконструкцию железнодорожного вокзала и автовокзала на станции Отдых. Победителем тендера на проведение строительных работ стала компания “Мастерок”, которая успешно провела реконструкцию лагеря отдыха для детей завода “Медтехника” и построила плавательный бассейн в нашем городе.

Работы по реконструкции вокзалов начались в сентябре и продлятся до февраля следующего года. В связи с проведением строительных работ доводим до сведения пассажиров, что пешеходный мост через железнодорожные пути закрыт, с платформ электричек на автовокзал можно попасть только через проезд Сапунова. Хотя это и длинный путь (идти неторопливым шагом придется около 15 минут), но безопасный. Призываем всех взрослых пресекать попытки детей пролезать под платформами и перебегать через железнодорожные пути.

Замечание 1. Выполнением этого задания должно стать расписание передвижения учащихся класса: сбор около ж/д касс, время отправления на электричке и т.д. При составлении расписания необходимо учитывать, что ожидается плохая погода, т.е. необходимо сократить ожидание и электрички, и автобуса, и регистрации в гостинице.

Замечание 2. Группа экспертов может принимать расписание без объяснений и выносить свое суждение о выполненном задании. Можно усложнить задание и потребовать обосновать свои предложения.

Умение сделать вывод о нацеленности имеющейся информации на решение конкретной проблемы

Пример 2. Задание “Составляем доклад”. Предложите учащимся заранее подобранные источники (5–7 небольших статей) по той теме, которую вы сейчас изучаете на уроках информатики и по которой требуется сделать 15-минутный доклад. Кроме этих источников, вы должны представить школьникам название доклада и его план. Задание состоит в следующем:

- 1) учащиеся должны оценить, достаточно ли представленной информации для подготовки доклада на заданную тему;
- 2) они должны указать по каждому пункту плана, из какого источника необходимо (или лучше) взять информацию (можно попросить указывать номер абзаца);
- 3) в случае если некоторый пункт плана раскрывается в нескольких источниках, учащийся должен обосновать, почему он отдал предпочтение конкретному источнику;
- 4) в случае если для конкретного пункта плана недостаточно информации, учащийся должен это указать и обосновать свой вывод.

Умение обосновать свои выводы

Умение обосновывать свои выводы трудноформализуемо. В разных ситуациях человек привлекает различные доводы для обоснования своих выводов. Например, в задании “Экскурсия” группой “ищущих решение” может быть предложен такой вариант:

9.20 — Сбор на ж/д вокзале
 9.43 — Отправление электрички от ст. Славное
 13.10 — Отправление автобуса до г. Чудесное
 13.45 — Прибытие в г. Чудесное
 14.00 — Регистрация в гостинице
 17.00 — Обзорная автобусная экскурсия по городу

Обосновать это расписание учащиеся могут по таким критериям:

- учитывается безопасность передвижения в районе строительных работ (достаточно времени для перехода от электрички до автобуса);
- учитывается ситуация, что в гостинице на регистрацию будет очередь (достаточно времени для регистрации в гостинице);
- достаточно времени для отдыха перед обзорной экскурсией.

Другая группа может предложить, например, такой вариант:

11.45 — Сбор на ж/д вокзале
 12.09 — Отправление электрички от ст. Славное
 16.05 — Отправление автобуса до г. Чудесное
 16.40 — Прибытие в г. Чудесное
 16.50 — Регистрация в гостинице
 17.00 — Обзорная автобусная экскурсия по городу

А в качестве обоснования указать: время сбора назначено на не слишком раннее время, можно выспаться в выходные. Заметим, что это не самый удачный вариант расписания.

Возможен и третий вариант с минимальным временем ожидания автобуса.

Для выбора одного (наилучшего) варианта следует сформулировать, какие требования мы выдвигаем к расписанию первого дня экскурсии.

Умение сбалансированно осветить вопрос при наличии противоречивой информации

При формировании этого умения необходимо обратить внимание учащихся на то, что если по одному и тому же вопросу существуют различные, быть может, противоречивые точки зрения, но которые представлены достоверной, полной, объективной (с точки зрения учащегося) информацией, то при освещении данного вопроса необходимо указать все точки зрения с обязательной ссылкой на источник (название источника, фамилия и имя автора, если они указаны, год издания и т.д.). Как правило, в такой ситуации от учащегося не требуется высказывать свое мнение, главное — нельзя игнорировать какие-то взгляды на проблему/вопрос, даже если они расходятся с мнением создающего новую информацию. Такая ситуация возникает при создании презентаций, написании рефератов на тему о современной жизни общества, развитии своего города, использовании природных ресурсов и т.п. В реферате или презентации учащийся должен осветить различные, часто противоречивые точки зрения, что является показателем сформированности умения “сбалансированное освещение вопроса”.

Структурирование созданной информации с целью повышения убедительности выводов

Это умение также является общеучебным, и формироваться оно должно на всех уроках. Формировать вы его можете наиболее эффективно при подготовке учащимися рефератов и докладов. Можно сказать, что в основе этого умения лежит способность *свертывать информацию* [2]. В данном случае результатом свертывания информации будет план изложения нового материала. Необходимость повышения убедительности информации возникает или в случае ее сложности из-за новизны, или в случае нечеткого изложения. В ситуации, когда учащийся осознал, что его освещение вопроса неубедительно, он должен задать себе вопросы: Что непонятно во вновь созданной информации? Почему непонятна информация? Что я хотел изложить (сказать)? Ответом на эти вопросы должен

быть пересмотр или уточнение плана (хотя часто оказывается, что ученик сдает реферат без какого-либо плана).

Замечание. Работу по оценке презентаций или рефератов советуем начинать с проверки плана. При отсутствии плана работа не должна приниматься. *Создавать новую информацию без плана нельзя!*

Приведем план-схему реферата [2] с примерным распределением по объему между разделами, с советами о характере используемой информации при работе над тем или иным разделом (табл. 2). Отметим, что данная схема применима и при создании презентаций.

План-схема

1. Вступление.
2. История вопроса.
3. Основное содержание.

- 3.1. Уточнение темы с обоснованием;
- 3.2. Современное состояние разработки вопроса или наиболее распространенное представление о каком-либо предмете, явлении, проблеме (с упоминанием имен, дат, конкретных примеров);
- 3.3. Альтернативные точки зрения (также с упоминанием имен и, возможно, со своим отношением);
- 3.4. Результаты самостоятельного исследования, проведенных наблюдений, опытов.
4. Связь рассматриваемого вопроса с другими областями знаний, отраслями промышленности и т.д.
5. Заключение, выводы.
6. Приложения.
 - 6.1. Список используемой литературы;
 - 6.2. Таблицы, диаграммы, графики;
 - 6.3. Схемы.

Таблица 2

| Номер раздела | Доля в общем объеме | Характер используемых информационных источников | Советы учащимся |
|---------------|------------------------|--|--|
| 1 | ≈ 5% | Начать работу помогают иногда вроде бы не относящиеся к делу ассоциация, цитата, стихи и т.д. | Возможно личностное вступление, отражающее отношение к данному предмету или теме |
| 2 | ≈ 15% | Учебник, энциклопедии, предисловия или послесловия научно-познавательных статей, электронных публикаций | Без исторического экскурса не обойтись, но слишком увлекаться не нужно |
| 3 | ≈ 50% | Энциклопедии последних 10 лет публикаций (например, “Аванта+”, электронная энциклопедия “Большая энциклопедия Кирилла и Мефодия” и т.д.), учебные и научно-познавательные книги, статьи с сайтов, заслуживающих доверия, статьи из научно-популярных журналов также последних 10 лет | Главные усилия и основное время нужно посвятить этой части работы. Уточнение темы обязательно будет связано с некоторым самоограничением. Лучше сконцентрироваться и глубже разобраться в одном-двух аспектах выбранной проблемы, чем утонуть во многих. Широту работы обеспечит “исторический” раздел (2) и следующий (4). Попробуйте разобраться в разных подходах, разных гипотезах или теориях выбранной вами проблемы |
| 4 | ≈ 10% | Современные энциклопедии, научно-познавательные книги, сайты | |
| 5 | ≈ 20% | | Кроме краткого резюме о проделанной по составлению реферата работе, нужно остановиться на прикладном значении рассматриваемого вопроса |
| 6 | Без ограничения объема | | Для оформления списка используемой литературы необходимо придерживаться правил библиографического описания информационных источников. Материалы Приложения могут отчасти повторять доказательную часть реферата, но этот материал будет более детальным, подробным |

Пример тестового задания “Вода в организме человека”

Время на выполнение — 15 мин.

Проверяемая компетентность: создание, интеграция

Полный сценарий. Вы взялись оформить химическую газету по теме “Вода в организме человека”.

Структура газеты:

| Вода в организме человека | | |
|--|--|--|
| Содержание воды в человеческом организме |  | Функции воды в человеческом организме |
|  | Недостаток воды в человеческом организме |  |

Информация, подобранная вашими одноклассниками:

1. Вода для человеческого организма — это второе по значимости вещество после кислорода.

2. Вода имеет большое значение для физической деятельности организма человека, выполняя такие функции, как транспортировка веществ, сохранение постоянства внутренней среды и теплообмен.

3. Вода помогает функционировать почти каждой части человеческого тела.

4. Тело ребенка от рождения до годовалого возраста содержит 80–85% воды. При достижении возраста 18 лет содержание воды уменьшается до 65–70%, а в старости — до 25%.

5. Без пищи человек может прожить 2–3 месяца, а без воды погибает через неделю.

6. Уменьшение количества воды в организме всего лишь на 1,5% вызывает сильную жажду, плохое самочувствие, сонливость, замедление движений, тошноту, иногда покраснение кожи.

7. В обычных условиях организм человека адаптируется к окружающим условиям, и баланс воды поддерживается как бы “сам собой”: захотел пить — попил.

8. Если в организме содержание влаги снижается на 6–10%, это чревато головной болью, одышкой, отсутствием слюноотделения, потерей способности двигаться и нарушением логического мышле-

ния. Основные рубрики в газете уже определены и имеют название. Ваши одноклассники нашли много интересной информации, и вся она кажется важной, подходящей к любому разделу. Заполните соответствующие рубрики газеты логически грамотно, не используя при этом ненужную, на ваш взгляд, информацию.

9. Сейчас люди используют 54% доступной пресной воды, причем две трети уходит на нужды сельского хозяйства.

10. Многие ученые считают, что человеческая жизнь в известной степени представляет собой “борьбу за воду”.

11. Вода — индикатор старения.

12. Вода содержится во всех органах человека. Мозг — это 75% воды. Кровь — это 92% воды. Кости — это 22% воды. Мускулы — это 75% воды, в них находится около половины всей воды тела. Стекловидное тело глаза — это 99% воды.

13. Вода регулирует температуру тела, несет питательные вещества и кислород ко всем клеткам тела. Вода требуется для дыхания, увлажняет кислород для дыхания, защищает и буферизирует жизненно важные органы, помогает преобразовывать пищу в энергию, помогает питательным веществам усваиваться органами, выводит отходы.

14. Вода является распространенным, простым и в то же время самым сложным и таинственным веществом на Земле.

15. Вода считается священной основой жизни, и с ней связаны важнейшие ритуалы практически всех

религий — например, крещение, очищение святой водой, омовение в водах.

16. От качества воды зависит качество нашей жизни. Хотя запасы пресной воды на Земле исчисляются миллионами кубических метров, далеко не всякая вода полезна для человека.

Краткий сценарий


Выбери соответствующие предложения и перетащи их в нужные рубрики газеты. Затем переставь

предложения так, чтобы получились логически грамотные сообщения.

Сценарий работы учащегося

- Прочитать сценарий и вводные инструкции;
- Прочитать и перетащить нужные сообщения по рубрикам, ненужные сообщения оставить неиспользованными;
- Переставить предложения в каждой рубрике, чтобы в каждом разделе получился логически связанный текст.

Ожидаемый (верный) результат выполнения задания

| Вода в организме человека | | |
|---|--|---|
| <p>Содержание воды в человеческом организме 1, 4, 12</p> |  | <p>Функции воды в человеческом организме 3, 2, 13</p> |
|  | <p>Недостаток воды в человеческом организме 5, 6, 8</p> |  |

Оценка сложности выполняемых действий

| Наблюдаемое действие | Сложность действия | Оценка действия в зависимости от выполненных работ |
|----------------------|--------------------|---|
| Интеграция | Высокий | Исключил всю несоответствующую информацию. Не выбраны 7, 9, 10, 11, 14, 15, 16 |
| | Средний | Выбрал одно-два лишних сообщения или не выбрал одно-два нужных сообщения |
| | Низкий | Все остальные случаи |
| Создание | Высокий | Во всех рубриках отображена нужная информация и расставлена в логической последовательности: 1, 4, 12; 5, 6, 8; 3, 2, 13 |
| | Средний | Во всех рубриках отображена нужная информация, но в одной из рубрик информация не расставлена в логически правильной последовательности (номера те же, но другой порядок) |
| | Низкий | Все остальные случаи |

Вопросы и задания

1. Ведется ли в вашей школе проектная работа? Если да, то какие педагогические приемы вы используете для формирования умения грамотного представления итоговой информации по проекту (презентация, реферат и т.д.)?

2. На основе прочитанных лекций подготовьте для администрации школы справку о проекте «Оценка ИКТ-компетентности девятиклассников».

Литература

1. Собкин В.С., Евстигнеева Ю.М. Подросток: виртуальность и социальная реальность. По материалам социологического исследования. Труды по социологии образования. Т. VI. Выпуск X. М.: Центр социологии образования РАО, 2001.
2. Голубева Е.И. О работе с книгой: Подсказки для школьников. М., 2006.
3. Реферат всему голова: Рекомендации по написанию рефератов для старшеклассников // Открытый урок, № 6, 1997.

Уважаемые слушатели курсов повышения квалификации!

Для получения зачета по пройденной части курса вам необходимо выполнить контрольную работу № 2. Работа должна быть отправлена по адресу: ул. Киевская, д. 24, Москва, 121165, Педагогический университет “Первое сентября” не позднее 15 января 2007 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” (напоминаем, что вы также можете воспользоваться разделом “Консультации” на сайте Педуниверситета <http://edu.1september.ru>). 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 лекций 5—7 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 10 баллов.

КОНТРОЛЬНАЯ РАБОТА № 2

по курсу И.Н. Фалиной, В.Ф. Бурмакиной

“Как готовиться к тестированию по проверке ИКТ-компетентности школьников”

1. Составьте список учебников и учебных пособий, который вы рекомендуете своим коллегам для использования при подготовке темы “Свойства информации”. Как называется деятельность по составлению такого списка в терминологии наших лекций (исходя из определения ИКТ-компетентности)?

(2 балла)

2. Заполните нижеприведенную таблицу:

| Вид деятельности (умение) | Основная цель, ради которой эта деятельность выполняется | Результат деятельности |
|-----------------------------|---|---|
| Определение (идентификация) | | |
| Доступ (поиск) | Выбор стратегии поиска и формирование информационного запроса | |
| Управление | | |
| Интеграция | | |
| Оценка | | Список информационных ресурсов, подходящих для решения нашей задачи |
| Создание | | |

Как называется деятельность, которую вы выполнили при заполнении таблицы? (3 балла)

3. Составьте как минимум два задания на формирование умений “интеграция”, “оценка” или “создание”. 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: | <input type="text"/> |
| Имя: | <input type="text"/> |
| Отчество: | <input type="text"/> |
| Идентификатор: | <input type="text"/> (указан в вашей персональной карточке) |
| КОММЕНТАРИИ | |
| | |

Примерные ответы на профильные билеты

Е.А. ЕРЕМИН, А.П. ШЕСТАКОВ,
г. Пермь

Продолжение. См. № 19, 20, 21, 22/2006

Билет № 7

1. Типы данных. Структуры данных. Обработка массивов. Итеративные и рекурсивные алгоритмы обработки массивов. Многомерные массивы

Обсуждая билет № 5, мы уже говорили о типах данных, их роли в алгоритме, а также о том, что дает компьютеру “знание” типа конкретной величины. Мы также говорили, что существуют простые и сложные типы данных. Простые типы чаще всего используются для хранения рабочих величин, но могут быть также аргументами или результатами в несложных алгоритмах. Тем не менее для представления данных в реальных задачах возможностей простых данных обычно недостаточно. В самом деле, в типичном случае о человеке необходимо хранить фамилию, имя и отчество (текстовые данные), год рождения (число), пол и семейное положение (одно из значений, выбираемых из фиксированного набора), факты наличия или отсутствия определенных признаков, например, обладания недвижимостью (логические данные) и т.д. В результате для объединения совокупности всех относящихся к одному реальному объекту данных (компонентов) требуется возможность создания из них единой сложной структуры.

Примечание. Стоит отметить, что если к сложному набору данных, состоящему из отдельных компонентов (полей), добавить методы их обработки, то получится автономная конструкция еще более высокого уровня — *объект* (см. билет № 6).

Необходимость в обработке на компьютере сложных видов данных также непосредственно вытекает из цикличности большинства применяемых на практике алгоритмов. Свойство массовости (см. билет № 4) требует, чтобы алгоритм реализовывался для обработки максимально большого количества данных, а не для какого-то одного конкретного значения¹. Следовательно, компьютер в подавляющем большинстве случаев производит многократную обработку множества данных по одному и тому же алгоритму (например, просматривает информацию о каждом человеке, отбирая по определенному признаку только тех, кто нужен согласно запросу). В результате в языке программирования необходимо предусмотреть средства, которые позволяют компактно представлять программы такого рода обработки: описать действия над одним из типичных данных, а затем циклически многократно повторять эти действия. Нетрудно по-

нять, что над формальной совокупностью простых данных (набором простых переменных с разными обозначениями) такую процедуру построить трудно — для этого лучше подойдет одна величина, определенным образом организованная (например, массив).

Как неявно следует из дальнейшей формулировки вопроса, в данном билете требуется рассказывать именно о сложных типах данных (в некоторых языках, подобных Си, подобные конструкции данных принято называть *структурами*).

Современные языки программирования позволяют создавать весьма разнообразные сложные структуры данных, причем для этого можно объединять как несколько простых, так и другие сложные данные. Самым распространенным сложным типом является **массив**, соединяющий в себе набор данных одного типа, например, массив из целых чисел или массив из логических величин. Кстати, конструкция “массив из массивов” также является вполне допустимой, о чем мы будем говорить несколько позже.

Для сложного типа в обязательном порядке требуется предварительно описать его состав. Это логически понятно, поскольку для конкретной структуры данных компоненты выбирает пользователь, и компьютер в принципе не в состоянии предугадать их набор; следовательно, никакое самое дружественное программное обеспечение не в состоянии автоматизировать данную операцию. В то же время сложные структуры данных требуют больших объемов памяти, так что и с этой точки зрения компьютеру требуется заранее знать состав компонентов структуры (для распределения памяти под *каждый* ее компонент).

Как уже отмечалось ранее, наиболее простой из сложных структур данных в памяти компьютера является массив. Поэтому мы будем рассматривать компьютерную обработку сложных данных именно на его примере.

Массив — это единая структура, компоненты которой имеют один и тот же тип. Обращение к отдельным элементам массива производится путем указания общего имени массива и определенной информации, характеризующей позицию требуемого элемента внутри массива. Последняя имеет смысл некоторой координаты (или, может быть, нескольких координат) и называется **индексом** (индексами). Отсюда видно, что массивы естествен-

¹ Действительно, для однократного вычисления по единственной не слишком сложной формуле обычно пользуются калькулятором, а не пишут программу.

ным образом делятся по количеству индексов на **одномерные** (с одним индексом), **двумерные** и, что допускается в большинстве языков, **многомерные** массивы (с большим числом индексов). Для удобства изложения начнем рассмотрение с одномерных массивов.

Описание одномерного массива в различных языках программирования выглядит несколько по-разному, но всегда содержит информацию об устройстве индекса и типе элементов. Например:

`m1: array [1..10] of real;` — Паскаль
`float m2[10];` — Си, Java и ряд других языков программирования
`dim m3(9) as variant;` — VBA (Visual Basic for Applications)

Во всех приведенных примерах в той или иной форме присутствует информация об индексах массива² и типе его компонентов³. Видно, что, несмотря на различную форму записи, ее смысл во всех языках программирования примерно один и тот же, что легко объясняется общностью целей описаний массивов.

Подчеркнем, что хотя в одних языках (Basic, Си) индексы, определяющие положение элемента массива, всегда числовые, в других (Паскаль, Ада) допускаются и другие “счетные” типы (более точно их называть *порядковыми*). Хорошим примером порядковых данных может служить символьный тип CHAR, упорядоченный в соответствии с принятым в компьютере алфавитом (см. билет № 21).

Следует четко различать значения *элементов* массива и значения их *индексов*, обеспечивающие доступ к этим элементам. Можно провести весьма глубокую аналогию, сопоставив эту пару понятий с человеком и его почтовым адресом, который позволяет доставить необходимую информацию нужному лицу.

При решении задач на обработку массивов очень важно постоянно контролировать корректность значений индексов. Особенно актуальной эта проблема становится в том случае, если значения индекса задаются не в явном виде, а вычисляются тем или иным способом. Рассмотрим в качестве примера следующий фрагмент программы. Пусть в описании сказано, что массив Q состоит из элементов с номерами от 1 до 5. Пусть далее в некотором месте программы находится оператор присваивания `Q[t+2] := 0`. Тогда в случае `t > 3` нулевое значение будет сохранено вне массива, точнее говоря, в то место памяти, где *находился бы* элемент под номером `t+2`, *если бы* он существовал. Так что в результате выполнения данного некорректного присвоения в лучшем случае будет “испорчено” значение какой-либо другой переменной, а в худшем — сама исполняемая программа. В любом случае найти такую ошибку крайне сложно:

² В Паскале указаны граничные значения индекса (точнее говоря, тип данных *ограничение*, частным случаем которого как раз является диапазон целых чисел), в Си — количество элементов, в VBA — максимальное значение индекса (минимальное, как правило, равно 0, что упрощает расчеты с индексами, но есть оператор его установки в 1).

³ real, float и variant соответственно.

для этого потребуется большое внимание и глубокое понимание логики работы программы.

Примечание. Возможно, у некоторых читателей возник вопрос: неужели компьютер не в состоянии проконтролировать “попадание” значения индекса в “разрешенный” диапазон? Разумеется, в состоянии, вот только контроль этот увеличивает длину программы и замедляет ее работу. Именно поэтому во многих компиляторах контроль индексов является отключаемым, причем ради эффективности итоговой программы по умолчанию контроль как раз бывает выключен (именно так работает система программирования Turbo Pascal фирмы Borland).

Введение сложного типа “массив” позволяет компактно описывать решение широкого круга важных для практики задач. К типовым задачам обработки массива относятся:

- заполнение его элементов по определенному закону;
- нахождение некоторого характерного элемента (например, максимального или первого нулевого) и, может быть, его положения;
- поиск элементов, удовлетворяющих определенному условию (или подсчет их количества);
- определение некоторых характеристик массива (сумма, среднее значение, наличие упорядоченности или симметрии);
- сортировка массива;
- перенос данных из одного массива в другой, разделение массива на части, объединение массивов — и некоторые другие. Подчеркнем, что к этим, казалось бы, весьма абстрактным упражнениям сводится огромное множество практических задач: от обслуживания каталогов дисков до обработки результатов соревнований.

В качестве примера рассмотрим задачу поиска в массиве положения максимального элемента, которое определяется числовым значением его индекса. Программа решения этой стандартной для массивов задачи на языке Паскаль может быть реализована следующим образом.

```
PROGRAM ind_max(INPUT, OUTPUT);
CONST n = 10;
m: ARRAY [1..n] OF INTEGER =
      (8, 10, 9, 4, 7, 2, 5, 6, 3, 1);
VAR k, im: INTEGER;
BEGIN im := 1;
      FOR k := 2 TO n DO
        IF m[k] > m[im] THEN im := k;
      WRITELN('max=', im)
END.
```

Программа настолько проста и традиционна, что не требует особых комментариев. Отметим только, что для простоты отладки⁴ элементы массива не вводятся с клавиатуры, а инициализируются входящими в текст программы значениями с помощью механизма так называемых *типизированных констант*. Разумеется, вместо этого для ввода может быть написан и стандартный цикл `FOR k := 1 TO n DO READLN(m[k])`.

⁴ У автора, как у практика, большое сочувствие вызывает предлагаемая ученикам в образцах к билетам задача “написать и отладить программу ввода и сортировки... массива из 20 элементов”.

Существует большое количество способов организации обработки массивов. Помимо рассмотренной выше традиционной реализации, можно дополнительно выделить **итерационный** и **рекурсивный** методы. Опишем подробнее названные технологии обработки массивов на примере классической задачи сортировки массива.

Примечание. В том, что такой сложности задание соответствует уровню проведения выпускного экзамена, свидетельствуют задачи под номером 3, предлагаемые в качестве образцов в билетах № 12 базового и № 17 профильного уровня (см. [1] или [2]).

Пример итеративной обработки массива

Рассмотрим итерационный способ сортировки элементов одномерного числового массива в порядке возрастания, суть которого заключается в следующем. Просматривая массив, будем менять местами его соседние элементы в тех случаях, когда они нарушают требуемый порядок, иными словами, больший элемент оказывается в положении с меньшим индексом. Очевидно, что каждая такая перестановка “соседей” с позиций сортировки улучшает ситуацию в массиве, но одного “прохода вдоль массива” скорее всего будет недостаточно. Максимальное число просмотров равняется количеству элементов массива без единицы, что достигается при наиболее неудачной начальной расстановке (когда самое маленькое число стоит последним). Тем не менее вполне возможны ситуации, когда требуемый порядок по указанному алгоритму удастся получить за меньшее число проходов; в частности, в предельном случае, когда массив уже упорядочен, в этом можно убедиться единичным просмотром.

Каждый просмотр массива, приближающий к решению поставленной задачи (причем результат обработки всегда служит начальным состоянием для нового просмотра), в компьютерной литературе называют *итерацией*. В словаре [3], например, дается такое определение данного термина: **итерация** — это “повторение пошагового процесса, когда результат предыдущего шага (шагов) используется для получения результата следующего шага”.

Вообще говоря, любой цикл обработки массивов есть итерация. Правда, как правило, итерационными алгоритмами чаще всего называют такие, в которых количество повторений заранее неизвестно. Наш алгоритм сортировки является именно таким.

Механизм сортировки описанным методом настолько широко известен, что здесь мы приведем только возможное решение на Паскале и совсем краткие пояснения к нему.

```
PROGRAM sort_iter(INPUT, OUTPUT);
CONST n = 10;
m: ARRAY [1..n] OF INTEGER =
      (10, 9, 8, 7, 6, 5, 4, 3, 2, 1);
VAR i, p: INTEGER; c: BOOLEAN;
BEGIN REPEAT c := FALSE;
      FOR i := 1 TO n - 1 DO
        IF m[i] > m[i + 1] THEN
```

```
      BEGIN p := m[i]; m[i] := m[i + 1];
        m[i + 1] := p;
        c := TRUE
      END;
      FOR i := 1 TO n DO
        WRITE(m[i], ' '); WRITELN;
      UNTIL NOT c;
END.
```

В программе в виде типизированной константы задан массив, который с точки зрения сортировки по возрастанию представляет наибольшую трудность. Итерационный процесс сортировки регулируется переменной *c*, которая имеет смысл наличия произведенных во время итерации перестановок: когда *c* после выполнения очередной итерации ложно, перестановок не было и сортировку можно прекратить. Первый из циклов FOR обеспечивает проход по массиву и обмен “неправильно стоящих” соседних элементов местами, а второй — служит для контрольного вывода на экран результатов каждой итерации сортировки.

Заметим, что в билете № 5 в связи с рассмотрением физической задачи о расчете распределения температуры в стержне также был подробно описан пример итерационного алгоритма обработки одномерного массива.

Примеры рекурсивной обработки массива

Рекурсия — это способ решения задачи, при котором производится последовательное ее сведение к аналогичной, но более простой. К рекурсии в принципе можно свести любой циклический алгоритм, но не наоборот: известным примером может служить алгоритм закраски на экране произвольной области, который элементарно реализуется рекурсивным путем, зато решение этой задачи в виде циклов весьма затруднительно.

Таким образом, *любая* задача по обработке массива может быть решена с помощью рекурсии. Мы возьмем в качестве примера ту же самую задачу сортировки массива, которая рассматривалась выше для итерационного метода. Попутно окажется, что нам потребуются написать нахождение максимума, что также можно сделать рекурсивно⁵. Следовательно, в итоге у нас будет сразу два примера рекурсивных алгоритмов.

Рассмотрим возможную реализацию программы.

```
PROGRAM sort_rekurs(INPUT, OUTPUT);
CONST n = 10;
m: ARRAY [1..n] OF INTEGER =
      (10, 9, 8, 7, 6, 5, 4, 3, 2, 1);
FUNCTION ind_max(k: INTEGER): INTEGER;
VAR w: INTEGER;
BEGIN IF k > 1 THEN BEGIN w := ind_max(k - 1);
      IF m[k] < m[w] THEN ind_max := w
        ELSE ind_max := k
      END
      ELSE ind_max := 1
END;
```

⁵ Авторы рекомендуют провести сравнение рекурсивного поиска положения максимума с написанным традиционным способом программой *ind_max*, приведенной выше.

```

PROCEDURE sort2(p: INTEGER);
VAR im, c: INTEGER;
BEGIN im := ind_max(p);
      c := m[im]; m[im] := m[p]; m[p] := c;
      FOR c := 1 TO n DO WRITE(m[c], ' '); Writeln;
      IF p > 2 THEN sort2(p - 1)
END;
BEGIN sort2(n)
END.

```

Описание начнем с рассмотрения рекурсивной функции `ind_max` для нахождения индекса максимального элемента. Ее рекурсивная логика может быть сформулирована следующим образом. Решение задачи для массива из n элементов весьма трудоемко, поэтому его постепенно сводим к более простому: для $n - 1$, $n - 2$ и т.д. элементов (см. вызов `ind_max(k - 1)`, который и является рекурсивным вызовом функцией самой себя). Так поступаем до тех пор, пока ответ не станет тривиальным: в массиве из одного элемента индекс максимума равен индексу этого единственного элемента (в нашем случае 1). Далее производится “обратный ход” — зная текущее значение `ind_max`, соответствующее ему значение максимума сравниваем с текущим элементом массива, после чего индекс наибольшего из сравниваемых элементов запоминаем. Таким образом, отрабатывается последовательная цепочка рекурсивных вызовов и без всякого цикла организуется полный перебор всех элементов массива.

Примечание. Понимание механизма вызова процедур⁶ делает возможность реализации их “самовызова” абсолютно очевидной; даже просто зная о факте разрешения вызова из одной процедуры другой, можно предсказать существование рекурсивности. Главная же проблема рекурсии заключается в том, чтобы организованный рассматриваемым способом процесс каким-то образом завершился.

Теперь, рассматривая функцию `ind_max` как единое целое и не обращая внимания на ее внутреннюю логику, обсудим второй рекурсивный алгоритм — сортировку массива, описанную в рекурсивной процедуре `sort2`. Ее логика работы такова. В массиве размерности n находится максимальный элемент, и он меняется местами с последним. В результате самый последний элемент массива займет свое окончательное место и останется рассортировать массив уже меньшего размера, т.е. $n - 1$. Аналогичным образом задача последовательно решается для все меньшего и меньшего массива, пока не останется массив из одного элемента, в котором, естественно, сортировка уже не требуется: условие $p > 2$ предотвращает вызов `sort2(1)` и процесс рекурсии благополучно завершается.

⁶ Многие методисты охотно включают в программу школьного курса информатики изучение стека и других динамических структур данных; при этом ни один из них не упоминает, что именно этот самый стек, аппаратным образом организованный, лежит в основе механизма реализации вложенных процедур (и даже задачи им параметров!).

Примечание. Имеющийся в `sort2` цикл вывода массива на экран играет вспомогательную роль. Читатели при желании могут без труда самостоятельно оформить его в виде рекурсивной процедуры (печатаем первый элемент, а затем вызываем процедуру вывода оставшегося массива!), реализовав тем самым чисто рекурсивную программу без единого цикла.

Основная программа состоит из единственной строки `sort2(n)`, которая запускает процесс сортировки, начиная с полного массива.

Примечание для учителей. Вполне возможно, что приведенное решение задачи для некоторых школ покажется слишком сложным. В этом случае вполне можно ограничиться рекурсивным решением ее части — нахождением положения максимума в массиве. Вообще сам факт наличия вопроса о рекурсии⁷ в тексте официальных “министерских” билетов для выпускных экзаменов в пору, когда слово “программирование” для многих методистов стало едва ли не ругательным, весьма наглядно демонстрирует тот идейный разброд, который внесла в содержание курса информатики энергичная борьба за наполнение школьного курса прикладными офисными технологиями!

Все обсуждавшиеся выше примеры рассматривали одномерные массивы данных. Они являются наиболее простыми и, кроме того, естественным образом согласуются с принципами организации памяти компьютера: согласно классическим идеям фоннеймановской архитектуры, память — это линейный массив пронумерованных ячеек. Следовательно, остается лишь получить формулу пересчета значений индексов в адреса ОЗУ и с элементами такого массива можно легко работать.

В математике, физике и некоторых других науках тем не менее широко используются и **многомерные** массивы. Поэтому такие структуры не могли не появиться в языках программирования. Фактически есть два способа определения многомерных массивов: непосредственный и создание сложной структуры. Мы ограничимся обсуждением двумерных массивов; обобщение для большей размерности делается аналогично. На языке Паскаль, например, эти два способа описания массивов выглядят так:

```

m1: ARRAY [1..3, 1..5] OF REAL;
m2: ARRAY [1..3] OF ARRAY [1..5] OF REAL;

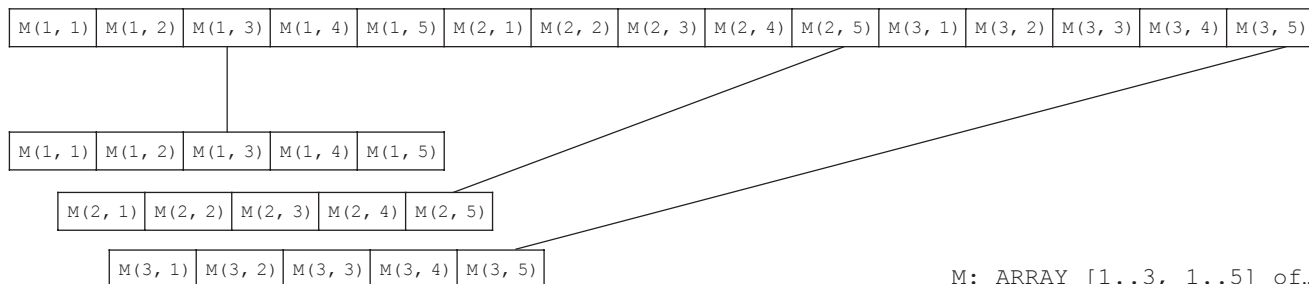
```

Примечание. Обращение к элементам массивов `m1` и `m2` отличаются: они записываются как `m1[i, j]` и `m2[i][j]` соответственно.

В некоторых простых языках программирования типа Basic используется только традиционная математическая (“матричная”) форма представления массивов, а в языках с более развитыми средствами создания данных, в частности Паскале, допустимы обе формы.

Подчеркнем, что если в языке программирования для индексов допустимы не только целочисленные типы (в Паскале в том числе), то индексы многомерных массивов не обязательно имеют одинаковый тип: например,

⁷ Рекурсию можно считать понятием третьего(!) уровня сложности в программировании: алгоритмические структуры — автоматные процедуры и функции — особенности их “самовызова”.



один индекс может быть целочисленным, а другой являться символом.

С математической точки зрения для увеличения размерности массива достаточно формально приписать еще один индекс. Однако для реализации такой структуры в памяти компьютера потребуется как-то приспособить многомерную структуру данных к хранению в одномерной памяти. “За исключением языка Fortran, все языки хранят двумерные массивы как последовательности строк... Такое размещение вполне естественно, поскольку сохраняет идентичность двумерного массива и массива массивов”. [4] Приведенный рисунок поясняет идею формирования двумерного массива данных в “линейном” ОЗУ компьютера.

Учитывая, что все элементы массива однородны (имеют один и тот же тип и, следовательно, занимают одинаковое место в памяти), получить формулу пересчета индексов в требуемый адрес ОЗУ является несложной задачей. И хотя школьных знаний для этого вполне достаточно, мы не будем сейчас этим заниматься. Заметим лишь, что в 32-разрядных процессорах Intel специально предусмотрены аппаратные методы адресации, в кото-

рые заложены такие формулы для данных стандартной длины (1, 2, 4 и даже 8 байт), что существенно облегчает для программиста (или для компилятора) доступ к элементам массивов. Заинтересовавшиеся читатели могут познакомиться с формулами для масштабированной базово-индексной адресации (одномерные массивы) и для аналогичного метода со смещением (двумерные), например, по книге [5].

Литература

1. Примеры задач к билетам. Информатика, 2006, № 6, с. 3–16.
2. Примеры задач к билетам. Информатика и образование, 2006, № 3, с. 24–30.
3. Фридланд А.Я. Информатика и компьютерные технологии: Основные термины: Толковый словарь / А.Я. Фридланд, Л.С. Ханамирова, И.А. Фридланд. М.: ООО “Издательство Астрель”: ООО “Издательство АСТ”, 2003, 272 с.
4. Бен-Ари М. Языки программирования. Практический сравнительный анализ. М.: Мир, 2000, 366 с.
5. Гук М. Процессоры Intel: от 8086 до Pentium II. СПб.: Питер, 1997, 224 с.

2. Изображение на бумажном носителе состоит из нескольких частей. Отсканировать части изображения и объединить их в одно растровое изображение. Отретушировать получившееся изображение и сохранить его в файле.

Решение данной задачи (после сканирования частей изображения) может быть выполнено в любом растровом графическом редакторе (в частности, в редакторе Paint). Для получения качественного итогового изобра-

жения рекомендуется воспользоваться редактором Adobe Photoshop.

Не будем здесь приводить детального описания решения задачи.

3. Определить информационный объем переданного сообщения за определенный период времени при заданной пропускной способности канала.

Пример. Модем передает сообщения со скоростью 14 400 бит в секунду. Изображение какого размера (в формате без сжатия) может передать модем за три минуты постоянной работы, если используется палитра из 65 тысяч цветов?

Решение. Предположим для определенности, что палитра составляет $65\,536 = 2^{16}$ цветов. Тогда для кодирования информации об одной точке требуется 2 байта. $14\,400 \text{ бит/с} = 1800 \text{ байт/с}$. За 3 мин. $= 180 \text{ с}$ будет передано $180 \times 1800 = 324\,000 \text{ байт}$ ($\approx 316,4 \text{ Кб}$) изображения или информация о 162 000 точек изображения.

Варианты заданий

1. Информация по электронной почте через модем передается со скоростью 9600 бит/с. Сколько времени понадобится, чтобы передать по электронной почте субботний выпуск газеты “Комсомольская правда”, если ее объем 3 условных печатных листа (1 условный печатный лист газеты с иллюстрациями $\approx 512 \text{ Кб}$)?

2. В течение урока 12 учеников пишут диктант в 10 000 символов. Оцените, сколько минут понадобится, чтобы переслать поочередно работы всех учеников на ПК учителя при скорости пересылки 9600 бит/с?

Азы информатики.

Выходим в Интернет

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

Продолжение. См. № 17–22/2006

Как работает Интернет

Содержание этого блока публикации:

- Академия. Формат IP-пакета. Формат IP-адреса.

Таблицы маршрутизации

- FAQ
- Полезные ссылки
- На уроке у Нины Дмитриевны (этот раздел будет опубликован в следующем номере)

Академия. Формат IP-пакета и Формат IP-адреса

В этом разделе приводится подробная информация о форматах IP-пакета и IP-адреса согласно документу RFC 791. RFC — это *Request For Comments*, запрос комментариев. Так по традиции называют все стандарты Интернета.

Формат IP-пакета

IP-пакет кроме данных, передаваемых по сети, содержит управляющую информацию (заголовок). Структура IP-пакета показана на рис. 5.1.

| | | | | |
|-------------------------------------|-----------------------------|-------------------------|---------------------------------|-----------------------------|
| Версия (4 бита) | Длина заголовка (4 бита) | Тип службы (8 битов) | Длина данных (16 битов) | |
| Идентификация (16 битов) | | | Флаги (3 бита) | Смещение пакета (13 бит) |
| Время жизни (8 битов) | Протокол (8 битов) | | Контрольная сумма (16 битов) | |
| IP-адрес отправителя (32 бита) | | | | |
| IP-адрес получателя (32 бита) | | | | |
| Параметры IP (может быть пустым) | | | Заполнение | |
| Данные | | | | |

Рис. 5.1. Структура IP-пакета

Ниже приводится описание отдельных полей IP-заголовка.

Версия

Это поле определяет текущую версию протокола IP, по которому работает передающая станция. В настоящее время используется версия 4 и идет работа над версией 6, в которой IP-адрес будет иметь длину не 32 бита, как в версии 4, а 128 бит, что позволит существенно расширить количество компьютеров, подключенных к Интернету.

Длина заголовка

Длину IP-заголовка необходимо указывать, так как она может быть разной в зависимости от длины поля *Параметры IP*. Длина заголовка задается количеством 32-битных блоков (слов).

Тип службы

Содержит указание на тип маршрута для данного пакета.

Такое указание позволяет правильно выбрать режим передачи данных по сети. Например, для службы IRC (чаты), работающей в режиме реального времени, необходима низкая задержка и высокая пропускная способность. Для службы FTP (передача файлов) подобные условия не требуются.

Поле *Тип службы* состоит из следующих частей:

- приоритет,
- задержка,
- пропускная способность,
- надежность.

Длина данных

Длина данных (длина пакета без заголовка). Под это поле отведено 16 бит, значит, данные в пакете могут иметь длину до 65 535 байт.

Идентификация

Показывает, какие пакеты относятся к одному и тому же сообщению. В конечном пункте эта информация необходима, чтобы пакеты разных сообщений не перепутались.

Флаги

Содержит специальную пометку для последнего пакета исходного сообщения.

Кроме того, это поле содержит бит отказа от дальнейшей фрагментации. Если маршрутизатор получает пакет, который он должен разделить на части для успешного перенаправления в сеть, не допускающую длинные пакеты, а бит отказа от фрагментации установлен, то пакет будет отброшен, а передающей станции будет послано сообщение об ошибке.

Смещение пакета

В Читальном зале говорилось, что каждый пакет нумеруется, чтобы сообщение в конечном пункте было правильно собрано.

На самом деле пакеты содержат не номера, а смещение в байтах относительно начала сообщения. Это смещение и записывается в этом поле.

Пусть исходное сообщение длиной в 3000 байт разбито на 3 пакета по 1000 байт в каждом. Тогда смещением первого пакета будет число 0, второго — 1001, третьего — 2001 (рис. 5.2).

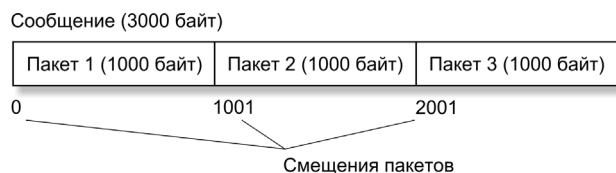


Рис. 5.2. Сообщение поделено на три пакета

В двоичном виде эти смещения будут записаны так:

| Номер пакета | Смещение |
|--------------|---------------|
| 1 | 0000000000000 |
| 2 | 0001111101001 |
| 3 | 0011111010001 |

Время жизни

Определяет интервал времени, в течение которого пакету позволено находиться в сети.

При прохождении через каждый маршрутизатор значение этого поля будет уменьшаться. Маршрутизатор, который уменьшит это поле до нуля, отбросит пакет и проинформирует об этом отправителя.

Протокол

В этом поле указывается, какой протокол должен обработать данный пакет при получении на станции назначения (обычно протокол TCP).

Контрольная сумма

Служит для определения пакетов, поврежденных при передаче. Пакет поврежден, если заново вычисленная контрольная сумма не совпадает со значением, записанным в этом поле. Поврежденные пакеты отбрасываются, а отправителю высылается соответствующее уведомление.

На каждой промежуточной станции меняется значение поля *Время жизни* и, возможно, поля *Параметры IP*, поэтому значение поля *Контрольная сумма* каждая станция для неповрежденных пакетов вычисляет заново.

IP-адрес отправителя

В этом поле записывается 32-битный IP-адрес отправителя сообщения. Адрес используется для послышки уведомлений в стартовый пункт послышки сообщения.

IP-адрес получателя

В этом поле записывается 32-битный IP-адрес получателя сообщения. Определяет станцию назначения пакета.

Параметры IP

В этом поле (переменной длины) маршрутизаторы делают отметки о прохождении пакета (трассировка маршрута). Кроме того, в этом поле записываются параметры безопасности передачи.

Заполнение

Поле не содержит никакой полезной информации. Оно вводится для выравнивания заголовка пакета на 32-битную границу (по соглашению длина заголовка пакета должна быть кратна 32 битам).

Формат IP-адреса

В 4-й версии протокола IP под адрес сетевого узла отводится 32 бита. Формат этого адреса можно представить в виде, представленном на рис. 5.3.

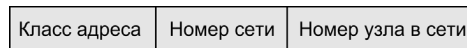


Рис. 5.3. Структура IP-адреса

Между этими тремя полями нет фиксированной границы. Для точного определения узла всегда используется полный адрес. Принятая запись адреса в виде четырех десятичных чисел w.x.y.z тем более не отражает представленное на рис. 5.3 деление адреса.

IP-адреса делятся на 5 классов: А, В, С, D и E. Классы позволяют организовать различное количество сетей и узлов в них.

Классы А, В и С используются для представления адресов сетей и узлов в них.

Класс D — это специальный тип адреса, используемый для групповой рассылки.

Класс E зарезервирован для экспериментальных целей.

На рис. 5.4 показаны двоичные форматы адресов для каждого класса.

Класс А (126 сетей, 16 777 214 узлов в каждой сети)

| | | |
|---|-------------------|--------------------|
| 0 | 7 бит номера сети | 24 бит номера узла |
|---|-------------------|--------------------|

Класс В (16 382 сети, 65 534 узлов в каждой сети)

| | | |
|----|--------------------|--------------------|
| 10 | 14 бит номера сети | 16 бит номера узла |
|----|--------------------|--------------------|

Класс С (2 097 150 сетей, 254 узлов в каждой сети)

| | | |
|-----|--------------------|-------------------|
| 110 | 21 бит номера сети | 8 бит номера узла |
|-----|--------------------|-------------------|

Класс D

| | | |
|------|---|--|
| 1110 | Групповой адрес в диапазоне 224.0.0.0 — 239.255.255.255 | |
|------|---|--|

Класс E

| | | |
|-------|--|--|
| 11110 | Класс E зарезервирован для экспериментальных целей | |
|-------|--|--|

Рис. 5.4. Структура IP-адреса по классам

Класс А определяется по первому биту адреса: если этот бит равен 0, то это адрес класса А.

Класс А определяет 50% доступного адресного пространства. В самом деле, под код класса отводится один бит, а так как бит может принимать два значения, то без него адресов будет в два раза меньше.

Класс адреса В определяется по первым двум битам: их значения должны быть 1 и 0.

В силу того, что из собственно адресации выпадают первые два бита, класс В определяет 25% доступного адресного пространства (число адресов без 2 битов уменьшается в 4 раза).

Класс В — самый популярный класс адресов, и в настоящее время адреса этого класса практически исчерпаны.

Класс С определяется по первым трем битам адреса: 110. Он определяет 12,5% всех адресов в сети (число адресов без 3 бит уменьшается в 8 раз).

Класс А дает много узлов, но мало сетей. Класс С дает много сетей, но мало узлов. Сбалансированный вариант предоставляет класс В, но его ресурсы на сегодняшний день практически исчерпаны.

Среди всех IP-адресов имеются несколько зарезервированных под специальные нужды:

| IP-адрес | Значение |
|--------------------------|-------------------------|
| Все нули | Данный узел сети |
| Номер сети все нули | Данная сеть |
| Все нули номер узла | Узел в данной сети |
| Все единицы | Все узлы данной сети |
| Номер сети все единицы | Все узлы указанной сети |
| 127.0.0.1 | “Петля” |

Адрес 127.0.0.1 предназначен для тестирования сетевого программного обеспечения в рамках одного компьютера.

Таблицы маршрутизации

Первый протокол, по которому работали сети, — протокол УИСП (*Unix to Unix Copy Protocol* — протокол копирования файлов от компьютера с операционной системой Unix на аналогичный компьютер).

Согласно этому протоколу в *таблице маршрутизации* расписывались связи между доменным адресом получателя сообщения и доменным адресом *соседнего* узла, в который нужно передать сообщение для продвижения его к пункту назначения.

Для уменьшения числа строк таблицы и ускорения поиска в ней в столбец “Адрес получателя” записывались не сами адреса, а шаблоны:

| Адрес получателя сообщения | Адрес соседнего узла, в который нужно передать сообщение |
|----------------------------|--|
| ... | ... |
| *.su | relcom.su |
| ... | ... |

Строка | *.su | relcom.su | программирует передачу всех сообщений, у которых адрес получателя имеет суффикс (окончание) “.su” (домен Советского Союза), в узел *relcom.su* (провайдер первой отечественной сети Релком на базе Курчатовского института).

Несмотря на шаблоны, последовательный поиск в такой таблице занимает много времени, что приводит к ощутимому замедлению работы сети.

В самом деле, нужно последовательно просматривать строки таблицы, сравнивая искомый адрес с шаблонами в первом столбце. В таблице на 1000 строк придется сделать 1000 таких сравнений, если шаблон искомого адреса расположен в последней строке.

Пусть N — число строк в таблице, тогда среднее число просмотров определяется, как $N/2$ (500 для нашего случая).

Для быстрого поиска в *упорядоченном* массиве существует метод *деления пополам* (дихотомия).

Суть метода: сравниваем объект поиска x со средним элементом t упорядоченного набора данных. Если $x = t$, поиск закончен. Если x меньше t , продолжаем поиск в первой половине набора, если больше — во второй.

Пусть, например, требуется найти число $x = 15$ в упорядоченном наборе (1, 5, 7, 15, 25, 40, 50, 65, 70).

Шаг 1. Средний элемент набора $t = 25$. Так как $x < t$, продолжаем поиск в первой половине набора (1, 5, 7, 15).

Шаг 2. “Средний” элемент в этом наборе $t = 7$. Так как $x > t$, продолжаем поиск во второй половине набора, который состоит теперь только из одного числа: (15).

Шаг 3. На этом шаге поиск завершается.

Известно, что среднее число просмотров при дихотомии оценивается сверху числом $\log_2 N$ (логарифм по основанию 2 от числа элементов в массиве поиска). Если таблица маршрутизации имеет 1000 строк, то по методу деления пополам придется выполнить не более 10 просмотров ($\log_2 1000 \approx 10$). Получаем ускорение по сравнению с методом последовательного (линейного) поиска в 50 раз (на таблице из 1000 строк).

Использовать дихотомию в таблице маршрутизации с доменными адресами нельзя: столбец поиска содержит данные, упорядочить которые невозможно из-за шаблонов.

В протоколе ТСП/IP в качестве сетевых адресов используются числа фиксированной длины (32 двоичных разряда) — IP-адреса.

Числовые строки таблицы маршрутизации легко содержать в упорядоченном виде, а это позволяет использовать для поиска быстрый метод дихотомии.

Но шаблоны в доменных таблицах маршрутизации УИСП существенно сокращали количество строк. Неужели в числовых таблицах ТСП/IP пришлось отказаться от задания в одной строке семейства адресов?

Конечно, нет. В TCP/IP аналогом доменного принципа адресации является принцип разделения адресов по подсетям (смотрите материал предыдущего раздела “Формат IP-адреса”).

IP-подсеть — несколько подряд идущих IP-адресов, двоичная запись которых имеет одинаковый префикс (номер сети) и различные суффиксы (номера узлов в этой подсети).

Таблица маршрутизации TCP/IP выглядит следующим образом:

| Адрес получателя пакета | Адрес соседнего узла, в который нужно передать пакет |
|-------------------------|--|
| ... | ... |
| Адрес подсети | IP-адрес |
| ... | ... |

FAQ

Приводятся ответы на следующие вопросы:

- Используется ли в Интернете коммутация линий?
- Говорят, что Интернет — это сеть сетей. Что это значит?
- Каков рекорд скорости передачи по сетям?
- Сетевая карта — это сетевой адаптер?
- Используется ли при работе с Интернетом через модем сетевая карта?
- Как работает сеть по протоколу УУСР?

Используется ли в Интернете коммутация линий?

Интернет работает по протоколу TCP/IP, для которого характерна передача сообщений методом коммутации пакетов.

Для переноса чувствительного к задержкам трафика (речь, видео) в Интернете используют передачу данных по *виртуальному каналу*.

Виртуальный канал прокладывается между узлами сети от начальной до конечной станции, а затем по нему перемещаются пакеты, относящиеся к одному и тому же сообщению.

Хотя передача в таком режиме напоминает обмен сообщениями по методу коммутации каналов (по заранее проложенному маршруту), она сохраняет принцип коммутации пакетов.

Физически канал не резервируется (не монополизируется), передача остается пакетной, кроме пакетов, передаваемых по виртуальному каналу, по его звеньям проходят пакеты, относящиеся к другим сообщениям.

На рис. 5.5 показан фрагмент сети, в которой передаются данные по виртуальному каналу S1—S2—S3—S4. Эта передача не исключает передачу пакетов, например, по пути S5—S2—S3—S6.

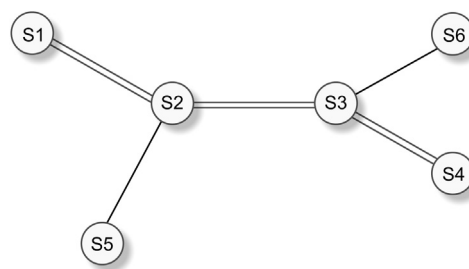


Рис. 5.5. Виртуальный канал S1—S2—S3—S4

Таким образом, по каналу S2—S3 будут попеременно проходить пакеты, относящиеся к двум разным сообщениям, что невозможно в сетях с коммутацией линий.

Говорят, что Интернет — это сеть сетей. Что это значит?

Интернет можно сравнить с мировой системой путей сообщения (автомобильные и железные дороги, авиационные и водные линии).

Каждый населенный пункт имеет дорожную сеть для собственных коммуникаций. Эти дорожные сети соединяются, образуя районную сеть. Районные сети объединяются в областные, областные — в государственные, государственные — в мировую транспортную сеть.

Аналогично объединяются и компьютерные сети. Отдельные учреждения или жители микрорайона создают локальные сети для своих внутренних коммуникаций. Эти сети соединяются, образуя региональные сети (городские, районные, областные). Соединение региональных сетей образует государственную сеть. Соединение государственных сетей — мировую глобальную сеть.

Конечно, к Интернету могут подсоединяться (и подсоединяются) отдельные компьютеры, но большая часть его пользователей распределена по региональным и корпоративным сетям.

Каков рекорд скорости передачи по сетям?

По кабельным сетям информация передается с самой большой скоростью. Скорости передачи в сетях с популярной технологией Ethernet представлены в таблице:

| Технология построения сети | Максимальная скорость передачи данных |
|----------------------------|---------------------------------------|
| Ethernet | 10 Мбит/с |
| Fast Ethernet | 100 Мбит/с |
| Gigabit Ethernet | 1000 Мбит/с |
| 10G Ethernet | 10 Гбит/с |

Скорости, которые приведены в таблице, привязаны к соответствующим стандартам. Однако принципиальные скоростные возможности передачи по кабелю существенно выше.

На сайте iXBT (<http://www.ixbt.com>) опубликованы две интересные статьи на эту тему.

NTT: скорость передачи данных по оптоволокну до 10 Тбит/с!

Новость от 28.04.2006. Адрес новости: www.ixbt.com/news/all/index.shtml?06/01/74.

Компания NTT Japan (*Nippon Telegraph and Telephone Corp.*) объявила о том, что ею успешно завершено тестирование новой технологии, позволяющей передавать данные на скоростях до 10 Тбит/с. В ходе эксперимента были использованы оптоволоконные кабели, соединяющие два города.

Однако не стоит преждевременно рассчитывать на мгновенное ускорение работы и значительный прорыв в достижении пропускной способности. В коммерческую эксплуатацию проект, как отмечается, поступит только к 2011 году... К тому же проект изначально разрабатывался и ведется в Японии, т.е. пройдет еще какое-то время, прежде чем технология будет лицензирована в Европе и запущена в эксплуатацию.

В ходе эксперимента передачи данных был использован 16-портовый роутер, на каждый порт приходилось 64 уплотненные линии связи с разными длинами волн. Мультиплексирование с разделением длины волны и уплотнением позволило передать по каждой линии поток данных в 640 Гбит, что в сумме и позволило добиться внушительных 10 Тбит/с.

Скорость Internet2 достигла отметки 8,8 Гбит/с!

Новость от 29.04.2006. Адрес новости: www.ixbt.com/news/all/index.shtml?06/01/84.

Как известно, проект Internet2 рассчитан на использование самых новых и совершенных на данный момент технологий передачи данных, которые в огромных (по сравнению с Internet1) объемах и на больших скоростях передаются между различными университетами и организациями мира, нуждающимися в высокоскоростных СПД.

Исследователи, работающие с Internet2, сообщили об установлении нового рекорда — скорости передачи данных, приближающейся к теоретическому максимуму (10 Гбит/с) — 8,8 Гбит/с. Рекорд был установлен сотрудниками Университета Токио, Университета в Амстердаме и другими их коллегами. В ходе эксперимента данные передавались из Токио через Сиэтл, Амстердам и Чикаго обратно в точку отправления. Скорость предыдущего рекорда, установленного в ноябре 2005 г., составляла 7,99 Гбит/с. И, несмотря на ощутимые успехи, операторы Internet2 сообщают о своих планах дальнейшего увеличения скорости передачи данных до 80 раз (!). Здесь будет уместно упомянуть о достигнутых успехах японских исследователей из NTT. На данный момент Internet2 использует оптоволокно Qwest Communications International. В новой Сети операторы будут использовать 10 “цветов”, т.е. 10 длин волн для передачи сигнала. Предполагается, что такой метод уплотнения потока (весьма схожий с японским решением) позволит достичь скоростей порядка 100 Гбит/с. Конечная же цель — применение 80 цветов. Это и даст 80-кратное увеличение скорости. Как сообщил Дуэлас Ван Ховелинг (Douglas Van Houweling), исполнительный директор проекта, сеть Internet2 с обновленными (до какой степени, не уточняется...) скоростными параметрами должна заработать к осени 2007 года. Если обычному пользователю сети Internet не совсем понятно, что даст ускорение передачи данных в Internet2, то ученые и исследователи по всему миру давно жаждут увеличенной пропускной способности. Скорейший ее рост вызван конечно же не желанием поставить очередной рекорд, а необходимостью. Так, например, астрономы по всему миру испытывают необходимость передачи огромных объемов

данных, связав воедино мощнейшие радиотелескопы (вполне возможно, и для задач поиска гравитационных волн).

Для сравнения: скорость чтения данных с жесткого диска компьютера не превышает 800 Мбит/с. Передача по каналу Gigabit Ethernet идет быстрее.

Передача непрерывного потока звуковой информации без компрессии требует пропускной способности канала около 1,5 Мбит/с, а передача полноэкранный видео — около 4,5 Мбит/с.

Скорость получения данных из Интернета определяется скоростью самого медленного канала из всех тех, по которым информация добирается со станции отправления. Если компьютер пользователя связан с провайдером через обычный аналоговый модем, то скорость приема данных из Интернета не может превышать 56 Кбит/с.

Сетевая карта — это сетевой адаптер?

Да, конечно.

Адаптер (от лат. *adapto* — приспособляю) в широком смысле — устройство, реализующее интерфейс (взаимодействие) между двумя разными средами. То есть адаптер — это соединительное устройство, переходник.

Адаптер в узком смысле — устройство связи компьютера с периферийными устройствами (например, видеокарта, или сетевая карта, или модем).

Сетевой адаптер совместно с программным обеспечением (драйвер карты и специальный сетевой модуль операционной системы) отвечает за прием данных из сети в компьютер пользователя и передачу данных в сеть.

Используется ли при работе с Интернетом через модем сетевая карта?

Модем — это сетевой адаптер, специализированный для передачи и приема информации по телефонным каналам связи.

Модем может быть выполнен в виде компьютерной карты, которая вставляется в слот материнской платы компьютера или в виде отдельного устройства, которое подключается к компьютеру (например, через порт COM или USB).

Для работы модема дополнительного сетевого адаптера не требуется, он сам является полноценным сетевым адаптером, специализированным для работы по телефонной линии.

Как работает сеть по протоколу UUCP?

UUCP (*Unix to Unix Copy Protocol*) — протокол, по которому работали компьютерные сети на этапе своего зарождения (70–80-е годы). Затем на смену пришел протокол TCP/IP, и сейчас протокол UUCP практически не используется.

По протоколу UUCP компьютеры, оснащенные операционными системами UNIX, могли обмени-

ваться информацией (почта, новости, файлы) через телефонные линии с помощью модема.

Для протокола УИСП характерна передача данных по принципу *коммутация данных и сеансная связь* между узлами сети.

Коммутация данных предполагала, что в каждом сетевом узле формировалась *таблица маршрутизации*, в которой расписывалась связь между адресом получателя сообщения и адресом *соседнего* узла, в который нужно передать сообщение для продвижения его к пункту назначения. При этом в таблице маршрутизации в столбце “Адрес получателя” записывались не сами адреса, а шаблоны:

| Адрес получателя | Адрес соседнего узла |
|------------------|----------------------|
| ... | ... |
| *.su | relcom.su |
| ... | ... |

Строка | *.su | relcom.su | программирует передачу всех сообщений, у которых адрес получателя имеет суффикс (окончание) “.su” (домен Советского Союза), в узел *relcom.su* (провайдер первой отечественной сети Релком на базе Курчатовского института).

Запись шаблонов вместо полных адресов позволяла существенно сократить таблицу маршрутизации, а значит, упростить ее заполнение и ускорить поиск в ней.

Коммутация данных отличается от коммутации пакетов тем, что сообщение передается *целиком* без деления на части-пакеты.

Сеансная связь предполагала, что для каждого узла составлялось расписание сеансов работы с соседними узлами сети.

Когда подходило время связи узла А с узлом В, модем А поднимал трубку и дозванивался до модема В. Модемы “представлялись” и “договаривались” о технических параметрах работы (важный параметр — скорость передачи данных). Затем модем А (автор звонка) начинал передавать данные, накопленные для узла В.

Когда узел А завершал передачу, ее начинал В, если у него имелись данные для А.

После обмена данными узлы разрывали связь до следующего сеанса.

Сеансы между узлами устанавливались примерно через каждый час, сообщения добирались до пункта назначения за сутки (иногда за двое суток, если промежуточных узлов было много).

Проблемы протокола УИСП: сеансная связь приводит к медленной работе сети, а коммутация данных — к монополизации канала связи (большой файл надолго закрывает канал связи для других узлов).

Еще одна проблема: таблица УИСП-маршрутизации содержала символические доменные адреса (к тому же разнородной длины). Поиск в такой таблице занимал много времени.

Проблемы протокола УИСП были решены протоколом TCP/IP.

В таблицах маршрутизации стали использовать упорядоченные по возрастанию числовые IP-адреса (адреса сетей, смотрите раздел Академии “Таблицы маршрутизации”).

Поиск в числовой упорядоченной таблице выполняется быстрым методом деления пополам.

Суть метода: сравниваем объект поиска x со средним элементом t упорядоченного набора данных. Если $x = t$, поиск закончен. Если x меньше t , продолжаем поиск в первой половине набора, если больше — во второй.

Пусть, например, требуется найти число $x = 15$ в упорядоченном наборе (1, 5, 7, 15, 25, 40, 50, 65, 70).

Шаг 1. Средний элемент набора $t = 25$. Так как $x < t$, продолжаем поиск в первой половине набора (1, 5, 7, 15).

Шаг 2. “Средний” элемент в этом наборе $t = 7$. Так как $x > t$, продолжаем поиск во второй половине набора, который состоит теперь только из одного числа: (15).

Шаг 3. На этом шаге поиск завершается.

По протоколу TCP/IP сообщения передаются разделенными на пакеты: это исключает монополизацию каналов связи одним сообщением.

Наконец, для магистральной части Интернета (на уровне соединения сетей) стали использовать выделенные кабельные и спутниковые каналы, устраняя тем самым сеансный режим передачи (магистральные узлы всегда на связи, прием/передача выполняется в любой момент времени).

Полезные ссылки

1. Сайт “Вебпланета”

www.webplanet.ru

Ежедневный электронный журнал. Разделы:

- *Интернет.* Электронная почта, браузеры, поисковые машины, спам, блоги...

- *Е-коммерция.* Интернет-магазины, онлайн-аукционы, платежные системы.

- *Безопасность.* Хакеры, вирусы, фишинг, дырки, черви, зомби и прочая нечисть.

- *Телекоммуникации.* Разборки провайдеров. Деж трафика. И немного о маршрутизаторах и оптоволокне.

- *Реклама и пиар.* О том, как нужно рекламировать товар в Интернете.

- *Технологии.* Все, что не поместилось в вышеперечисленные рубрики, но о чем все-таки очень хочется рассказать.

- *Интервью.* Прямая речь именитых особ.

- *Блицинтервью.* Прямая речь именитых особ на конкретную тему. Комментарии к статьям и новостям.

- *Мнения.* Поток сознания наших читателей.

- *Читальный зал.* Аналитические отчеты, долгосрочные прогнозы, описание тенденций. С графиками, таблицами, статистикой.

2. iXBT

ixb.com

Новости, статьи, руководства, конференции, интернет-магазин, гид по ценам.

Брянская областная командная олимпиада среди школьников по информационным и коммуникационным технологиям

А.А. ИЛЬЕНКО,
г. Брянск

Весной 2006 года состоялась Третья областная командная олимпиада среди школьников по информационным и коммуникационным технологиям, проводимая Департаментом общего и профессионального образования области совместно с Брянским региональным центром информатизации и компьютеризации.

Стало доброй традицией собираться в областном центре, чтобы в честной борьбе определить команду, лучше всех владеющую навыками работы с информационными и коммуникационными технологиями. Областной олимпиаде предшествовали районные и городские, победители которых и приехали в Брянск. В этом соревновании приняли участие 34 команды из 27 районов и городов области, а также команда из г. Ливны Орловской области, которая изъявила желание принять участие вне конкурса.

Особенностью нынешней олимпиады является выполнение заданий и общение между членами команды в режиме реального Интернета. Несмотря на значительно возросший по сравнению с прошлым годом уровень подготовленности участников, некоторые команды не смогли выполнить все задания олимпиады, т.к. не владели достаточными навыками работы в глобальной сети.

Результат закономерен — общеобразовательные учреждения, где информационные технологии занимают достойное место в образовательном процессе, представили наиболее подготовленные команды.

К такому формату проведения областной олимпиады по ИКТ мы пришли не сразу. Традиционно в области проводится олимпиада по программированию, но уже не первый год продолжается тенденция к снижению интереса как учеников, так и учителей информатики к данной олимпиаде. Мы не ставим целью анализировать причины создавшегося положения, но факт остается фактом и олимпиада по программированию становится уделом избранных школьников. Но это единицы среди тысяч наших учеников, которые с удовольствием изучают информатику в школе. И поэтому несколько лет назад перед нами встала задача нахождения путей вовлечения в олимпиадное движение массового школьника. Апробировались разные подходы в проведении олимпиады по ИКТ: и проведение теоретических туров с включением вопросов из всего курса информатики, и зрелищные командные соревнования на сцене, и

личная олимпиада школьников. В результате нескольких лет таких поисков мы пришли к следующей схеме проведения олимпиады по ИКТ.

Районный этап олимпиады

В зависимости от технических и иных возможностей района оргкомитет рассматривает следующие варианты проведения олимпиады.

Вариант 1

Олимпиада проводится в трех образовательных учреждениях, имеющих выход в Интернет. В каждом из этих учреждений работает один участник от команды (все капитаны собираются в одном месте). Для каждого члена команды организаторами заранее регистрируется адрес электронной почты, которые получают все участники команды перед началом работы.

В установленное время капитанам направляется письмо с заданием или адрес сайта, где находится задание, по электронной почте. Капитан должен оптимально распределить задания между членами команды. В ходе выполнения работы команда обменивается информацией по электронной почте (помогают друг другу, по необходимости обмениваются заданиями и т.д.). По окончании работы итоговые результаты аккумулируются у капитана и отправляются по электронной почте на адрес жюри или публикуются на указанном в задании сайте.

Время, отведенное на выполнение работы, — два астрономических часа.

Помимо ОС Windows 2000/XP, на каждом рабочем месте участника должен быть установлен необходимый набор программных средств: пакет Microsoft Office; растровый графический редактор Photoshop; архиватор WinRar.

За нарушение правил предлагается введение штрафных баллов, вплоть до отстранения команды от участия. Запрещается: общаться по электронной почте и другим средствам связи со всеми, кроме членов своей команды; выходить в форумы в Интернете; использовать для выполнения заданий другие программные средства. Контроль за выполнением правил осуществляют представители оргкомитета и жюри.

Допускается проведение олимпиады с использованием Брянской региональной сети "Метро" и иных

районных (городских) интрасетей. В этом случае задания по поиску в Интернете через поисковые системы заменяются организаторами на задания по поиску внутри конкретных ресурсов.

Вариант 2

Олимпиада проводится в одном образовательном учреждении, располагающем достаточным количеством компьютеров, объединенных в локальную сеть с функционирующей локальной почтой. Расположение участников осуществляется таким образом, чтобы максимально ограничить членов одной команды от непосредственного общения.

На каждом рабочем месте должен быть установлен необходимый набор программных средств (см. *Вариант 1*).

Задания по поиску в Интернете могут быть заменены организаторами на задания по поиску в программах — эмуляторах поисковых систем. При отсутствии таких систем допускается замена задания по поиску на предложенное организаторами задание олимпиадного характера (либо снятие этого задания и сокращение времени работы).

Задания для отборочного тура находятся в файле *zadanie.rar*, на сайте Брянского регионального центра информатизации и компьютеризации по адресу <http://book.debryansk.ru/events/olympiad2006/>.

При отсутствии необходимого количества компьютеров в одном образовательном учреждении рекомендуется проведение предварительных отборочных туров по своим текстам и правилам, позволяющим отобрать несколько лучших команд для районного тура.

Используя данные схемы, оргкомитеты на местах смогли оптимально выбрать способ проведения олимпиады и вовлечь в олимпиадное движение большое количество школьников.

Областной этап олимпиады

Вне зависимости от выбранного варианта на местах областной этап проводится согласно правилам, обозначенным в *Варианте 1*. Участники команды, победившей в I этапе, должны владеть навыками работы в сети Интернет (работа с электронной почтой, навигация в сети, поиск информации). Рекомендуется организовать для финалистов предварительные занятия в реальном Интернете на отработку указанных умений и навыков.

Другой вопрос, вставший перед организаторами, — сами олимпиадные задания. И если раньше, на этапе отработки механизмов проведения, задания были в основном базового уровня, то в этом году мы повысили сложность заданий. Количество заданий было избыточным, учитывая тот факт, что это командное соревнование и успех зависит от слаженной работы всей команды.

Многие задания были подсказаны методической литературой, в том числе газетой “Информатика”, открытыми публикациями в Интернете, но среди представленных много и авторских заданий.

В этом году в олимпиаде приняла участие команда наших орловских друзей, а в перспективе мы бы хотели видеть данную олимпиаду межобластной. В конце мая мы встречаемся в Орле у наших коллег на совместном семинаре и, думается, решим вопрос совместного проведения олимпиады по ИКТ. Мы открыты для всех и, если к нам захотят присоединиться и другие регионы, будем рады расширению круга общения. Тем более что выезжать, по большому счету, никуда не надо, нас свяжет Всемирная паутина!

Ниже приведены задания областного тура. С ними также можно познакомиться по адресу <http://book.debryansk.ru/events/olympiad2006/work/index.php>.

Задания областного тура олимпиады по ИКТ

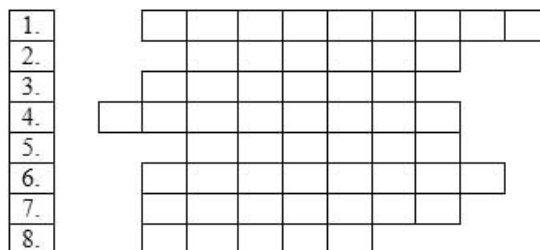
Организационное задание

Создать папку на рабочем столе под именем команды или района, куда в процессе выполнения заданий будет сохраняться вся информация.

Задание № 1 (20 баллов) *Этот человек...*

Откройте файл под именем *poisk.doc*. С помощью поиска в Интернете найдите фамилии людей, внесших заметный вклад в развитие информатики, заполните таблицу, находящуюся в этом файле (см. таблицу на с. 24).

Средствами Word в файле *poisk.doc* сделайте заготовку для кроссворда. Впишите результаты заглавными буквами в клетки кроссворда. Отыщите и выделите по вертикали получившееся ключевое слово.



Сохраните файл с ответами и кроссвордом в своей папке под именем *poisk.doc*, заархивируйте его и перешлите капитану.

Задание № 2 (10 баллов)

Создайте с помощью Word таблицу из 16 столбцов и 8 строчек. Заполните первый столбик указанными числами, записанными в разных системах счисления.

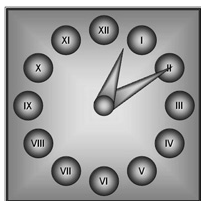
диться и скрывается за замком. Плавно цвета позднего заката меняются на цвета сумерек. Постепенно в небе начинают по очереди проявляться звезды, вслед за которыми проявляется луна.

Замок должен быть сделан средствами векторной графики MS Office как единый объект. На рисунках представлены сцены из сюжета. Обратите внимание на смены оттенков в окраске замка в зависимости от освещенности.

Презентацию сохраните в своей папке в файле под именем *zamok.ppt*, заархивируйте ее и перешлите капитану.

Задание № 4 (15 баллов)

Создайте слайд с изображением часов, расположенных строго в центре слайда. Запуск стрелок осуществляется сразу после запуска презентации, остановка — по нажатию кнопки “ДАЛЕЕ”. Условная минута равна 5 секундам, т.е. секундная стрелка совершает полный оборот за 5 секунд.



Движения секундной и минутной стрелок должны быть согласованы.

Презентацию сохраните в своей папке в файле под именем *clock.ppt*, заархивируйте ее и перешлите капитану.

Задание № 5 (10 баллов)

Средствами электронных таблиц решите старинную задачу.

Собирается баба на базар — на людей посмотреть да кое-что продать. Сколько надо взять бабе на базар для продажи животных гусей, уток и кур, чтобы выручить как можно больше денег, если она может взять товара не более 25 кг и известно, что:

- масса одной курицы — 1 кг, стоимость 60 рублей;
- масса одной утки — 1,5 кг, стоимость 100 рублей;
- масса одного гуся — 3 кг, стоимость 190 рублей.

Сохраните результат в своей папке в файле под именем *tovar.xls*, заархивируйте его и перешлите капитану.

Задание № 6 (45 баллов)

В архиве *spiski.rar* находятся файлы с результатами экзаменов по трем предметам: математика, обществознание и русский язык. Экзамены сдавали одни и те же люди, но некоторые люди могли не сдавать один или несколько экзаменов, некоторые сдавали все. Необходимо, используя пакет программ MS Office, достичь поставленных задач:

1. Определить Ф.И.О. всех, кто набрал максимальный балл по всем трем экзаменам (10 баллов).

2. Определить количество участников, набравших по математике 12, 11, 10, ..., 1, 0 баллов соответственно и 10, 9, 8, ..., 1, 0 по обществознанию и русскому языку (10 баллов).

3. Определить Ф.И.О. всех участников, которые набрали более 26 баллов за экзамены (15 баллов).

4. Определить Ф.И.О. всех, кто набрал максимальный балл в одном из экзаменов, а два других не сдавал вообще (20 баллов).

Сохраните результат в своей папке в файле под именем *exam.doc*, заархивируйте его и перешлите капитану.

Задания № 7, 8, 9 выполняются в редакторе растровой графики Adobe Photoshop.

Задание № 7 (10 баллов)

Создайте коллаж: второй цветок в файле *boikal.jpg* поместите в бокал. Обратите внимание, что часть цветка будет находиться в бокале.



Сохраните рисунок в своей папке в файле под именем *kollaj1.jpg*, оптимизировав его для пересылки по электронной почте, и перешлите капитану.

Задание № 8 (20 баллов)

Из двух файлов — *beaut.jpg* и *list.jpg*



beaut.jpg



list.jpg

создайте коллаж:



Сохраните рисунок в своей папке в файле под именем *kollaj2.jpg*, оптимизировав его для пересылки по электронной почте, и перешлите капитану.

Задание № 9 (15 баллов)

Восстановите старую фотографию, которая находится в архивном файле *foto.rar*, защищенном паролем. Пароль вы сможете получить, выполнив конкатенацию ключевых слов из заданий № 1 и 2. Сохраните рисунок в своей папке в файле под именем *foto.jpg*, оптимизировав его для пересылки по электронной почте, и перешлите капитану.

Задание № 10 (выполняется капитаном команды) (10 баллов)

В Блокноте создайте HTML-страничку, содержащую результат работы команды.

На странице должно быть отражено следующее. Заголовок окна — «Страница результатов команды <название команды>»

Цвет фона страницы — #FFCC33, основной цвет текста — #00660F, цвета гиперссылок — по умолчанию.

Строка 1. Название команды, оформленное заголовком 1-го уровня, цвет текста — #000066.

Строка 2. Название района — заголовок 2-го уровня.

Строка 3. «Члены команды» — заголовок 3-го уровня.

Ниже третьей строки идет перечисление членов команды в виде нумерованного списка, созданного средствами HTML.

Результаты выполненных командой заданий оформляются в таблице, как гиперссылки на архивные файлы.

Параметры таблицы: ширина — 50%; первый столбец — 30% от ширины таблицы; параметры для границы: толщина — 1 пиксель, цвет — #003300.

Капитан публикует все результаты работы команды на сервере по адресу <http://www.eduinfo.debryansk.ru/upload/> и отправляет по электронной почте на адрес webmaster@eduinfo.debryansk.ru письмо с сообщением об окончании работы команды.



«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)

I ВСЕРОССИЙСКАЯ ЗАОЧНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ ПО ИНФОРМАТИКЕ 2006/2007 УЧЕБНОГО ГОДА

Опыт проведения московской олимпиады по информатике на протяжении нескольких последних лет показал, что заочный тур — один из наиболее эффективных способов поиска талантливых школьников и подготовки их к очной олимпиаде. Также к нему проявили большой интерес школьники и учителя других регионов России.

В связи с этим было решено организовать и провести Всероссийскую заочную олимпиаду по информатике, взяв за основу модель заочного тура Московской региональной олимпиады.

Олимпиаду проводят Московский центр непрерывного математического образования, Московский институт открытого образования, Московский государственный университет им. М.В. Ломоносова. Олимпиада проводится с 1 октября 2006 г. по 10 января 2007 г. на сайте www.olympiads.ru.

Участвовать в олимпиаде могут все желающие, присоединившись к ней в любое время вплоть до **10 января 2007 года**. При этом в конкурсном зачете участвуют только школьники, все остальные (студенты, учителя и т.д.) могут участвовать в олимпиаде вне конкурса.

На сайте предложен набор задач различной сложности, а также предоставлена возможность проверки решений в режиме реального времени. Решением каждой задачи должен являться исходный текст программы на одном из допустимых языков программирования. Неработающие, а также частичные решения можно доделывать и сдавать повторно. По окончании олимпиады на сайте будут выложены решения задач и тесты, на которых проверялись решения.

В настоящий момент в системе автоматической проверки решений поддерживаются следующие языки: Free Pascal 2.0.4, GNU C 4.1.1, GNU C++ 4.1.1, Turbo Pascal 7.0, Borland C/C++ 3.1, QBasic 4.5, Borland Delphi 6.0, Java JDK 1.5.0 (жюри не гарантирует, что все задачи можно решить на Turbo Pascal, Borland C/C++, QBasic — программам на этих языках во время работы доступно около 500 Кб оперативной памяти).

Учитывать результаты Всероссийской заочной олимпиады по информатике уже собираются организаторы очных олимпиад Москвы, Санкт-Петербурга, Кирова, Ханты-Мансийского автономного округа и оргкомитет Летней компьютерной школы (подробная информация — на сайте олимпиады).

Мы предлагаем использовать результаты Всероссийской заочной олимпиады для приглашения успешно

выступивших школьников вашего региона на региональную олимпиаду по информатике. Другие способы отбора на региональную олимпиаду при этом также могут быть сохранены.

Что необходимо сделать, чтобы официально учесть результаты выступления школьников вашего региона в заочной олимпиаде?

- Оргкомитет вашей региональной олимпиады должен принять решение о том, что он собирается учитывать результаты Всероссийской заочной олимпиады (точные критерии могут быть объявлены позднее).

- Вы должны прислать письмо на имя председателя оргкомитета Всероссийской заочной олимпиады по информатике профессора А.Л. Семенова с указанием того, что ваш регион готов учитывать результаты Всероссийской заочной олимпиады по информатике, с указанием фамилии, имени, отчества, должности и контактных координат ответственного лица. Письмо может быть прислано по факсу (495) 291-65-01 или по e-mail zaoch@olympiads.ru. После получения письма информация об участии вашего региона в проведении Всероссийской заочной олимпиады по информатике будет опубликована на сайте олимпиады.

- По окончании олимпиады оргкомитет предоставит указанному ответственному лицу информацию о результатах участников вашего региона (с указанием фамилии, имени, класса, города, школы, e-mail и набранных баллов каждого участника). Оргкомитет передаст ту информацию, которую участники укажут при регистрации для участия в олимпиаде. Оргкомитет не несет ответственности за достоверность регистрационной информации.

Оргкомитет заочной олимпиады готов рассмотреть предложения о включении представителей вашего региона в оргкомитет и жюри.

В случае, если вам по каким-то причинам неудобен срок окончания олимпиады (например, ваша региональная олимпиада проводится в конце декабря), можно установить и опубликовать на сайте, что приглашение на вашу региональную олимпиаду происходит по текущим результатам на некоторую заранее определенную и объявленную дату (например, по состоянию на 23:59 15 декабря).

По всем вопросам, связанным с олимпиадой, пожалуйста, обращайтесь по электронной почте zaoch@olympiads.ru.

Игра "Сто к одному"

О.В. КОРОЛЕВА,

г. Нытва Пермского края

Предлагаю коллегам сценарий простой игры, которую можно провести в рамках предметной недели или в конце года, когда уже выставлены оценки. В игру "Сто к одному" с удовольствием играют ученики любого возраста. Поскольку правила основаны на принципах одноименной телевизионной игры, они знакомы большинству участников. Учителю нужно лишь оформить доску и провести небольшую подготовительную работу по сбору статистики ответов на вопросы игры. У нас в школе перед игрой всем проходящим по коридору школы (школьникам, учителям, родителям) задавались вопросы, связанные с предметом "информатика". Все ответы записывались.

Оформление доски:

| Простая игра | Двойная игра | Игра наоборот | СТО К ОДНОМУ |
|----------------------|----------------------|----------------------|--------------|
| <input type="text"/> | <input type="text"/> | <input type="text"/> | |
| <input type="text"/> | <input type="text"/> | <input type="text"/> | |
| <input type="text"/> | <input type="text"/> | <input type="text"/> | |
| <input type="text"/> | <input type="text"/> | <input type="text"/> | |
| <input type="text"/> | <input type="text"/> | <input type="text"/> | |

На доске расположены названия туров игры. Подготовленные ответы закрыты табличками.

В игре участвуют две команды.

К ведущему приглашаются по одному участнику от каждой команды для участия в **Простой игре**.

Ведущий задает вопрос: **Что можно сделать с информацией?**

Право ответить первым получает участник, чей хлопок в ладоши все услышали первым. Если участник дал один из популярных ответов, то этот ответ на доске открывается, команда при этом зарабатывает очки = количеству человек, давших при опросе такой же ответ.

Право ответа предоставляется той команде, чей участник дал наиболее популярный ответ. Если команда даст три неверных версии, то право на ответ и на заработанные очки может получить другая команда.

Ответы (здесь и далее приведена статистика ответов, полученная в нашей школе; у вас эти значения, наверное, будут другими):

- Обработать (26 человек)
- Прочсть (2)
- Передавать (10)
- Измерить (1)
- Хранить (7)

Следующая игра — **Двойная** (очки, полученные командой, умножаются на 2).

Ведущий задает вопрос: **Компьютер включен, но вдруг погас монитор. Почему?**

Ответы:

- Нет электроэнергии (11)
- Это заставка (3)
- Перегрузка (9)
- Выключили (2)
- Сломался (8)
- Отошел провод (1)
- Перегорел (5)

Затем проводится **Игра наоборот**. Задача команды — отгадать ответ, наименее популярный.

Ведущий задает вопрос: **Назовите одно из устройств ввода информации**

Право ответа предоставляется той команде, у которой очков меньше.

Ответы:

- Клавиатура (18 человек, 10 очков)
- Мышь (15, 20)
- Джойстик (6, 40)
- Трекбол (3, 80)
- Световое перо (2, 160)
- Микрофон (1, 320)

Подводятся итоги для того, чтобы выяснить, какая из команд участвует в большой игре.

Большая игра проводится следующим образом.

Один участник выходит за дверь в сопровождении участника из другой команды.

Его друг по команде отвечает на вопросы:

- **На чем можно печатать?**
- **Что такое винт?**
- **Какая кнопка есть на системном блоке?**
- **Курсор медленно передвигается по экрану. Почему?**
- **Что вредно компьютеру?**

Возвращается другой участник. Отвечает на эти же вопросы. Если дает ответ, который уже звучал, в течение 20 секунд ему нужно дать другой ответ.

Ответы:

- **На чем можно печатать?**
 - На клавиатуре (40 человек)
 - На бумаге (3)
 - На компьютере (1)
 - На печатной машинке (1)
- **Что такое винт?**
 - Жесткий диск (10)
 - Вентилятор (7)
 - Пропеллер (3)
 - Устройство для закручивания (2)
- **Какая кнопка есть на системном блоке?**
 - Вкл./выкл. (38) — Перезагрузка (13)
- **Курсор мыши медленно передвигается по экрану.**

Почему?

- Мышь плохая/старая (12)
- Грязный шар (9)
- Компьютер висит (3)
- Сломался компьютер (2)
- **Что вредно компьютеру?**
 - Пыль (11)
 - Вирусы (7)
 - Чистка (4)
 - Вода (4)
 - Поломка (1)
 - Зависание (1)
 - Вредно падать (1)
 - Внезапное выключение без завершения работы (5)

НАЧАЛКА

газета-клуб для всех,
кто учит информатике
маленьких детей



№ 16 (1-15 декабря)

Материалы конкурса "ТРИЗформашка-2006". Часть III

Н.Г. ИВАНОВА, М.А. ПЛАКСИН, О.Л. РУСАКОВА,
г. Пермь

Окончание. См. № 21, 22/2006

Мы завершаем публикацию материалов пермского конкурса по информатике для младших школьников. Материал получился очень объемным, но, надеемся, содержательное разнообразие заданий и тщательная детализация решений и правил оценивания помогут коллегам при организации мероприятий, подобных "ТРИЗформашке".

Ответы и правила оценивания (окончание)

15. Словарь

Ответы.

Задание 1. Составление словаря

| | | | |
|--|---|---|--|
| награда | | | наоборот |
| награда надвое адеться надолго наедине | назад наизнанку наизусть наискосок накануне | накидка накипь наклеить налево налетке | налим налог намерение нанимать наоборот |
| наотмашь | | | нарочно |
| наотмашь наперебой наперекор наперерез наперечет | напильник напиток наповал наподобие напомнить | направо например напрокат напротив напрямик | наравне нараспев нарасхват народ нарочно |
| наряд | | | невод |
| наряд насекомое население наскоро наспех | настурция натошак натура наука нафталин | начало начеку начерно начисто наяву | небосклон невежда невзгода невидимка невод |

Задание 2. Поиск слова “напротив”

| Слово | Буквы, которые надо читать |
|--------------|----------------------------|
| 1) наоборот | нао |
| 2) нарочно | нар |
| 3) напильник | напи |
| 4) направо | напра |
| 5) наравне | нар |
| 6) например | напри |
| 7) напрокат | напрок |
| 8) напротив | напротив |

Задание 3. Поиск слова “наугад”

| Слово | Буквы, которые надо читать |
|--------------|----------------------------|
| 1) наоборот | нао |
| 2) нарочно | нар |
| 3) невод | не |
| 4) настурция | нас |
| 5) начало | нач |
| 6) натошак | нат |
| 7) натура | нат |
| 8) наука | наук |

Поскольку слово “наука” должно стоять после слова “наугад”, можно сделать вывод, что слова “наугад” в словаре нет.

Баллы.

Задание 1. Составление словаря

За содержание — 22 балла:

- за алфавитный порядок — по 1 за колонку (5 подряд идущих слов). Максимум — 12.
- за верхние индексы — по 1 за слово. Максимум — 6.
- за расположение по колонкам — по 1 за разворот (4 колонки). Максимум — 4.

За ввод в ЭВМ — 1 балл за каждые 5 слов. Максимум — 13.

Всего — 35 баллов.

Задание 2. Поиск слова “напротив”

За правильный перечень слов и букв — 20. За каждую ошибку (неправильное слово, неправильную букву) — штраф 1 балл.

За ввод в ЭВМ — 1 балл за каждые 5 слов. При правильном ответе — 3.

Всего — 23.

Задание 3. За поиск слова “наугад”. — Аналогично.

Всего — 23.

Итого за все задание — 81 балл.

16. Информация о федеральных округах

Баллы.

Содержание и оформление оцениваются отдельно.

За содержание таблицы — максимум 90:

За название большого города и количество жителей в нем — по 1 баллу. Всего 6.

За федеральные округа — по 2 балла. Всего 6.

За название региона и его площадь — по 8 баллов. Всего 48.

За название второго города региона и количество жителей в нем — по 5. Всего 30.

За неправильную ячейку — 0 баллов.

Ответ.

| Самый большой город | Количество жителей в городе (тыс. чел.) | Федеральный округ | Самый большой по площади регион округа | Площадь региона (тыс. кв. км) | Второй по числу жителей город региона | Количество жителей в городе (тыс. чел.) |
|---------------------|---|-------------------|--|-------------------------------|---------------------------------------|---|
| Москва | 10 392 | Центральный | Тверская область | 84 | Ржев | 64 |
| Санкт-Петербург | 4624 | Северо-Западный | Архангельская область | 587 | Северодвинск | 199 |
| Новосибирск | 1413 | Сибирский | Красноярский край | 2340 | Норильск | 133 |

В таблице существуют зависимости: правые графы определяются левыми (неправильный выбор округа автоматически сделает неправильными все более правые графы). Участникам дается право на ошибку. А именно. Ошибки в первых трех графах отсекаются от ошибок в правых четырех графах. Это значит, что выбор города и округа оценивается отдельно от выбора региона и города в нем. Если неправильно выбран округ, то регион проверяется для того округа, который выбран учащимся. Например, если вместо Центрального округа выбран Южный, то в качестве самого крупного региона должна быть указана Волгоградская область, а вторым городом в ней — Волжский. Далее приводится необходимая информация по всем федеральным округам (самый большой регион округа, его площадь, второй по численности город и его население).

Оформление: числа располагаются посередине, выравниваются поразрядно. Все названия с большой буквы (здесь это естественно). За нарушение каждого из этих правил — штраф 1 балл.

За ввод в ЭВМ — 1 за каждые 5 слов. За полностью правильную таблицу — 5 баллов.

Итого: 95 баллов.

| Федеральный округ | Самый большой по площади регион | Площадь региона (тыс. кв. км) | Второй по числу жителей город региона | Количество жителей в городе (тыс. чел.) |
|-------------------|---------------------------------|-------------------------------|---------------------------------------|---|
| Центральный | Тверская область | 84,1 | Ржев | 64 |
| Северо-Западный | Архангельская область | 587,4 | Северодвинск | 199 |
| Южный | Волгоградская область | 113,9 | Волжский | 311 |
| Приволжский | Пермская область | 160,6 | Березники | 171 |
| Уральский | Тюменская область | 1435,2 | Сургут | 290 |
| Сибирский | Красноярский край | 2339,7 | Норильск | 133 |
| Дальневосточный | Республика Саха (Якутия) | 3103,2 | Нерюнгри | 75 |

17. Заполнение таблицы на основе блок-схемы

Баллы.

Содержание и оформление оцениваются отдельно.

За содержание — максимум 265 баллов:

За перечень округов по алфавиту — 2.

За название первой и последней графы — по 1. Всего — 2.

За название 3-й и 4-й графы — по 3. Всего 6.

В названии 4-й графы должно быть указание на то, что это *самый крупный* город округа (а не просто “Название города”). Если название просто “Название города” — 1 балл.

За название таблицы — 3.

В названии должны быть либо перечислены характеристики округов (количество регионов, самый крупный город), либо должны присутствовать слова типа “некоторая информация”, “краткие сведения” и т.п. Если название просто “Федеральные округа” — 1 балл.

За количество регионов в каждом округе — по 10. Всего — 70.

За каждый самый крупный город региона — по 25. Всего — 175.

За количество жителей в городе — по 1 баллу. Всего — 7.

За неправильную ячейку — 0 баллов.

За оформление — максимум 12 баллов:

Заголовок: в середине, с большой буквы, без точки — 3.

Заголовки граф: с большой буквы, без точки, именительный падеж, единственное число — 4.

Единицы измерения (“тыс. чел.”): они есть, в нужном месте (в заголовке графы), правильно выделены (в круглых скобках или через запятую) — 3.

Программа: выравнивание чисел (посредине поразрядно) — 2.

За невыполненные требования баллы не начисляются.

За ввод в ЭВМ — по 1 за каждые 5 слов. За содержимое таблицы — 5 баллов. Дополнительные баллы за заголовок таблицы и заголовки граф (в образце их будет 3).

Итого: 285 баллов.

Ответ. Табл. 5.

Таблица 5

Краткая информация о федеральных округах

| Федеральный округ | Вспомогательный столбец (см. алгоритм) | Количество регионов в округе | Самый крупный город округа | Население города (тыс. чел.) |
|-------------------|--|------------------------------|----------------------------|------------------------------|
| Дальневосточный | | 10 | Владивосток | 590 |
| Приволжский | | 15 | Нижний Новгород | 1297 |
| Северо-Западный | | 11 | Санкт-Петербург | 4624 |
| Сибирский | | 16 | Новосибирск | 1413 |
| Уральский | | 6 | Екатеринбург | 1287 |
| Центральный | | 18 | Москва | 10 392 |
| Южный | | 13 | Ростов-на-Дону | 1062 |

18*. Преобразование в таблицу информации о погоде

Баллы.

Ответ.

Вариант 1

Таблица 1

Погода

| Регион | Погодный параметр | | | |
|----------------------|--------------------|------|------------|------|
| | Температура, град. | | Осадки, мм | |
| | Январь | Июль | Январь | Июль |
| Красноярский край | -18 | +17 | 36 | 55 |
| Марий Эл | -10 | +20 | 26 | 79 |
| Московская область | -8 | +20 | 45 | 66 |
| Оренбургская область | -11 | +21 | 35 | 89 |
| Пермская область | -14 | +18 | 52 | 45 |
| Удмуртия | -13 | +19 | 41 | 61 |

* К сожалению, при публикации этого задания (на границе номеров 21 и 22) сбилась нумерация. И в № 22 это задание имеет номер 16. Это ошибка. Приносим читателям свои извинения.

Вариант 2

Таблица 1

Погода

| Регион | Месяц | | | |
|----------------------|--------------------|------------|--------------------|------------|
| | Январь | | Июль | |
| | Температура, град. | Осадки, мм | Температура, град. | Осадки, мм |
| Красноярский край | -18 | 36 | +17 | 55 |
| Марий Эл | -10 | 26 | +20 | 79 |
| Московская область | -8 | 45 | +20 | 66 |
| Оренбургская область | -11 | 35 | +21 | 89 |
| Пермская область | -14 | 52 | +18 | 45 |
| Удмуртия | -13 | 41 | +19 | 61 |

Оформление и содержание оцениваются отдельно.

Оформление: Всего — 38 баллов.

Табличный номер: Он есть, имеет вид “Таблица 1” (или “Таблица 2”, или “Таблица 3” и т.д.), справа вверху, с большой буквы, без точки, без знака номер — 6.

Заголовок: Он есть, правильный по содержанию (2 балла), в середине, с большой буквы, без точки — 6.

Головка — 17.

Расположена сверху, заголовки есть у всех граф (в том числе у боковика), выравнивание по верхней линии (для одноярусной головки этот балл не начисляется) — 3.

Все заголовки (погодные параметры и месяцы): с большой буквы, без точки, именительный падеж, единственное число — 4.

Количество ярусов в головке: три яруса — 10, два яруса — 7, один ярус — 0.

Головка одноярусная, но ячейка над боковиком поделена по диагонали. Над чертой — название класса объектов, расположенных в головке (по сути, заголовок верхнего яруса), под чертой — заголовок боковика — 3. (Вообще-то так делать не надо. Но будем считать, что это — неверно оформленный верхний ярус головки. И лучше иметь его оформленным неверно, чем не иметь вообще.)

Единицы измерения: Они есть, на нужном месте (2 балла), правильно выделены (через запятую или в круглых скобках) — 4.

Боковик: расположен слева, регионы упорядочены по алфавиту — 2.

Замечание. И заголовок графы, и заголовок строки должны однозначно идентифицировать обозначаемый объект или свойство. А это значит, повторяющихся значений ни в головке, ни в боковике быть не должно. Исходя из этих соображений, можно считать, что у таблицы с линейной структурой не может быть одновременно и головка, и боковик. Либо то, либо другое в зависимости от направления, в котором “вытянута” таблица (сверху вниз или справа налево).

Прографка: выравнивание чисел посередине поразрядно — 2.

Строк больше, чем граф — 1 (Требование не обязательно. Призовой балл).

За лишние (ошибочные) элементы оформления — штраф по 1 баллу за ошибку.

Содержание. Всего — 122 баллов.

Разделяются оценка правильности значений в таблице и оценка “степени структурирования” данных (линейная структура, двумерная структура).

Выглядит это так:

За структуру таблицы:

| | |
|-----------------------------|------|
| Степень структурированности | Балл |
| Линейная структура | 10 |
| Двумерная структура | 100 |

За правильность значений:

| Значение | Количество | Цена | Стоимость |
|------------------------------------|------------|------|-----------|
| Число в прографке | 24 | 1/4 | 6 |
| Регион | 6 | 1/2 | 3 |
| Заголовок боковика | 1 | 1 | 1 |
| Заголовок на нижнем ярусе головки | 4 | 1 | 4 |
| Заголовок на втором ярусе головки | 2 | 2 | 4 |
| Заголовок на третьем ярусе головки | 1 | 4 | 4 |
| Всего | | | 22 |

За неверную ячейку — 0. Пустая ячейка является неверной.

За лишнюю ячейку (в любой части таблицы) — штраф в 0,5.

Суммарная оценка таблицы — не менее 0.

Всего за содержание: 122.

Всего за оформление и содержание $38 + 122 = 160$ баллов.

За ввод в ЭВМ — 10.

Итого: 170 баллов.

19. Производство чириккукуков

Ответ состоит из таблицы и диаграммы.

Таблица 1

Производство чириккукуков 1999–2005 гг. (шт.)

| Продавец | Год | | | |
|--------------|------|------|------|------|
| | 1999 | 2001 | 2003 | 2005 |
| Траляля | 18 | 27 | 39 | 49 |
| Трулюлю | 12 | 36 | 51 | 35 |
| Тирлимбомбом | 20 | 37 | 60 | 56 |
| Всего | 50 | 100 | 150 | 140 |

Баллы за таблицу.

Оценивается отдельно оформление и содержание.

Оформление. Всего — 33 балла.

Табличный номер: Он есть, имеет вид “Таблица 1” (или “Таблица 2”, или “Таблица 3” и т.д.), справа вверху, с большой буквы, без точки, без знака номер — 6.

Заголовок: Он есть, правильный по содержанию (2 балла), в середине, с большой буквы, без точки — 6.

Головка — 14.

Расположена сверху, заголовки есть у всех граф (в том числе у боковика), выравнивание по верхней линии (для одноярусной головки этот балл не начисляется) — 3.

Все заголовки: с большой буквы, без точки, именительный падеж, единственное число — 4.

Количество ярусов в головке: два яруса — 7, один ярус — 0.

Головка одноярусная, но ячейка над боковиком поделена по диагонали. Над чертой — название класса объектов, расположенных в головке (по сути, заголовок верхнего яруса), под чертой — заголовок боковика — 3. (Вообще-то так делать не надо. Но будем считать, что это — неверно оформленный верхний ярус головки. И лучше иметь его оформленным неверно, чем не иметь вообще.)

Единицы измерения: Они есть, на нужном месте (2 балла), правильно выделены (через запятую или в круглых скобках) — 4.

Боковик: расположен слева — 1.

Замечание. И заголовок графы, и заголовок строки должны однозначно идентифицировать обозначаемый объект или свойство. А это значит, повторяющихся значений ни в головке, ни в боковике быть не должно. Исходя из этих соображений, можно считать, что у таблицы с линейной структурой не может быть одновременно и головка, и боковик. Либо то, либо другое в зависимости от направления, в котором “вытянута” таблица (сверху вниз или справа налево).

Прографка: выравнивание чисел посредине поразрядно — 2.

За лишние (ошибочные) элементы оформления — штраф по 1 баллу за ошибку.

Содержание. Всего — 96 баллов.

Разделяются оценка правильности значений в таблице и оценка “степени структурирования” данных (линейная структура, двумерная структура).

Выглядит это так:

За структуру таблицы:

| | |
|-----------------------------|------|
| Степень структурированности | Балл |
| Линейная структура | 4 |
| Двумерная структура | 40 |

За правильность значений:

| Значение | Количество | Цена | Стоимость |
|-----------------------------------|------------|------|-----------|
| Число в прографке | 16 | 3 | 48 |
| Продавец | 3 | 1/3 | 1 |
| Заголовок боковика | 1 | 1 | 1 |
| Заголовок на нижнем ярусе головки | 4 | 1 | 4 |
| Заголовок на втором ярусе головки | 2 | 1 | 2 |
| Всего | | | 56 |

За неверную ячейку — 0. Пустая ячейка является неверной.

За лишнюю ячейку (в любой части таблицы) — штраф в 0,5.

Суммарная оценка таблицы — не менее 0.

Числа в прографке оцениваются дорого, поскольку получены в результате сложных расчетов: надо взять сумму со столбиковой диаграммы, проценты — с круговых и провести перерасчет в абсолютные величины.

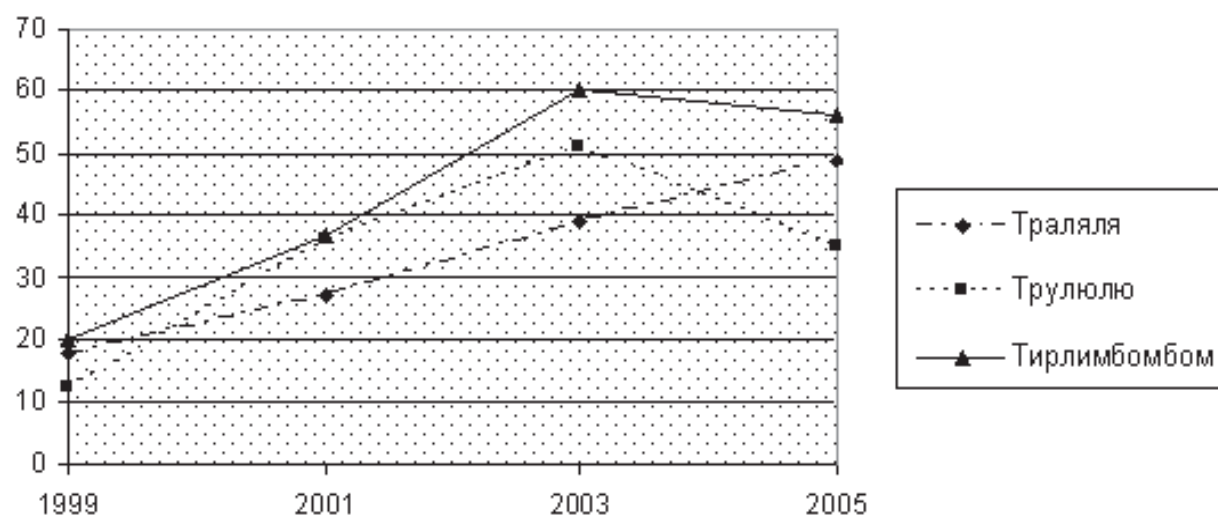
Всего за содержание: 96.

Всего за оформление и содержание $33 + 96 = 129$ баллов.

За ввод в ЭВМ — 10.

Итого за таблицу: 139 баллов.

Изменение производства чириккукуков (шт.)



Баллы за диаграмму.

Оценивается отдельно оформление и содержание.

Важным моментом является выбор правильного типа диаграммы для отображения динамики. Неверный выбор типа диаграммы обесценивает всю остальную работу. Поэтому при неверном выборе типа диаграммы от набранных баллов остается только 10%.

Содержание диаграммы (правильные данные в правильном порядке) — 48. (Отображаемых значений — 12. По 4 балла за каждое правильное значение.)

Оформление — 23 балла:

- заголовок диаграммы — 2;
- единицы измерения (они есть, на нужном месте, правильно выделены) — 3;
- легенда — 2;
- подписи на осях — 2;
- тип линий — 6 (должен быть свой тип для каждой линии);
- обозначение точек — 6 (должно быть свое обозначение для каждой линии);
- линии начинаются непосредственно от вертикальной оси (крайнее левое значение по горизонтальной оси совпадает с точкой пересечения с осью вертикальной) — 2.

Всего за оформление и содержание $48 + 23 = 71$ балл.

Внимание! Диаграмма должна быть обязательно линейная. Если тип диаграммы неверен (любая диаграмма, кроме линейной), полученную величину умножаем на 0,1.

За ввод в ЭВМ — 10.

Итого за диаграмму — 81 балл.

Всего за таблицу и диаграмму: $139 + 81 = 220$ баллов.

20. Исполнитель ТЕКСТ-ФЕКС-ПЕКС

Ответ и баллы.

Баллы начисляются:

- за составление алгоритма;
- за ввод в ЭВМ указания на положение курсора;
- за ввод алгоритма в ЭВМ;
- за ввод в ЭВМ текста песни.
- за определение позиции курсора;

Баллы за составление и ввод алгоритма приведены ниже в зависимости от алгоритма.

За определение позиции курсора — 1 балл. Указание может быть любым (словесное описание, значок на указуемой позиции).

За ввод указания на позицию курсора — по 1 за каждые 5 слов, не менее 1.

За ввод в ЭВМ текста песни — по 1 за куплет, всего — 5 баллов.

1-й вариант. Вложенные циклы. 8 команд. Курсор остается на две строки ниже последней строки текста.

```
ПОВТОРИТЬ 5 РАЗ НЦ
  ПОВТОРИТЬ 2 РАЗА НЦ
    ЗАПОМНИТЬ
    УДАЛИТЬ
    ВНИЗ
    ВСТАВИТЬ
```

```
КЦ
ВНИЗ
ВНИЗ
```

КЦ

Баллы. За алгоритм — 35 баллов.

За плохое оформление (отсутствие “лесенки”, маленькие буквы, запись нескольких команд в одной строке и пр.) — штраф до 5 баллов.

За ввод в ЭВМ алгоритма — 3 балла.

2-й вариант. Циклический. 11 команд. Курсор остается на две строки ниже последней строки текста.

```
ПОВТОРИТЬ 5 РАЗ НЦ
```

```
  ЗАПОМНИТЬ
  УДАЛИТЬ
  ВНИЗ
  ВСТАВИТЬ
  ЗАПОМНИТЬ
  УДАЛИТЬ
  ВНИЗ
  ВСТАВИТЬ
  ВНИЗ
  ВНИЗ
```

КЦ

Баллы. За алгоритм — 25 баллов.

За плохое оформление (отсутствие “лесенки”, маленькие буквы, запись нескольких команд в одной строке и пр.) — штраф до 5 баллов.

За ввод в ЭВМ алгоритма — 3 балла.

3-й вариант. Линейный. 48 команд. Курсор остается на последней строке текста.

Должен быть 4 раза выписан следующий фрагмент:

```
ЗАПОМНИТЬ
УДАЛИТЬ
ВНИЗ
ВСТАВИТЬ
ЗАПОМНИТЬ
УДАЛИТЬ
ВНИЗ
ВСТАВИТЬ
ВНИЗ
ВНИЗ
```

Затем то же самое, но без двух последних строк.

Баллы. За алгоритм — 5 баллов.

За плохое оформление (маленькие буквы, запись нескольких команд в одной строке и пр.) — штраф до 3 баллов.

За ввод в ЭВМ алгоритма — 10 баллов.

Если по предложенному в решении алгоритму не получить верного текста (в алгоритме есть логические ошибки), то оценка 0 баллов.

В мир информатики

82 (1—15 декабря)

*Газета для пытливых учеников
и их талантливых учителей*

Еще раз о программе Debug

Е.А. Еремин,
г. Пермь

Предисловие

В нашей газете была опубликована статья [1], посвященная простейшим приемам работы с программой-отладчиком Debug, некогда разработанной еще для операционной системы (ОС) MS-DOS, но по-прежнему входящей в состав ОС Windows. В статье были рассмотрены эксперименты с регистрами центрального процессора; в ней было продемонстрировано, как с помощью весьма несложных манипуляций можно понаблюдать за выполнением простейшей программы внутри современного компьютера типа IBM PC.

Хочется подчеркнуть, что возможности старой доброй программы Debug для образовательной демонстрации фундаментальных принципов работы компьютера значительно шире. И хотя существуют отладчики гораздо мощнее и с более современным интерфейсом, но простота использования и присутствие Debug в любой версии Windows позволяет считать его полезным в тех случаях, когда речь идет не о профессиональной отладке больших программ, а об изучении простейших закономерностей функционирования компьютера. В свете сформулированной позиции, редакция планирует опубликовать цикл статей, посвященных демонстрации тех или иных фундаментальных основ информатики и вычислительной техники с помощью данного отладчика. Предполагается, что материалы могут быть использованы как “пытливыми учениками”¹, которые интересуются указанной проблематикой, так и преподавателями различных учебных заведений. По мнению автора, некоторые наиболее простые эксперименты могут быть продемонстрированы даже на уроках базового курса информатики.

В этом выпуске публикуем первую статью цикла. Согласно традициям классического образования, начинать следует с теории. Как говаривал мудрый литературный персонаж Козьма Прутков, “бросая в

воду камешки, смотри на круги, ими образуемые; иначе такое бросание будет пустою забавою”. Поэтому первый эпизод знакомства с отладчиком будет посвящен ему самому. Если вы не верите, что “такое старье” заслуживает внимания, то вот вам “вопрос на засыпку”: как, по-вашему, должна работать программа Debug, отображая в столь подробном виде (см. [1]) содержимое многочисленных регистров микропроцессора, чтобы при этом не испортить значение ни в одном из них? Или еще: как отладчик умудряется останавливаться после каждой команды отлаживаемой программы при ее пошаговом исполнении?

1. Немного истории

Необходимость в отладочных программах существует с тех пор, как люди начали программировать. Не случайно поэтому уже в первой операционной системе для 8-разрядных микрокомпьютеров, CP/M (автор — Г.Килдэл, 1975 г.), существовало специальное отладочное средство, которое называлось DDT (*Dynamic Debugging Tool*). В одной из классических книг [2] по этой операционной системе роль отладчика охарактеризована весьма удачно. “Программа DDT позволяет пользователю загружать в оперативную память, просматривать, тестировать и изменять и отлаживать любую программу на языке ассемблера, представленную в машинном коде... Программа DDT работает в динамическом режиме, что позволяет пользователю запускать находящуюся в памяти отлаживаемую программу и отслеживать каждый шаг ее выполнения. С ее помощью можно также вносить небольшие изменения в существующие программы (такие, как текстовый процессор WordStar) или реассемблировать их с тем, чтобы понять, как они работают... Программа DDT представляет собой важнейшее инструментальное средство для программистов, использующих язык ассемблера. Для эффективного использования этой программы необходимо понимание принципов программирования на языке ассемблера”.

¹ См. логотип газеты-вкладки “В мир информатики”. — Ред.

Кроме DDT, существовал аналогичный отладчик SID (*Symbolic Instruction Debugger*) и его расширенная версия для процессора Z80² под названием ZSID.

С переходом к 16-разрядным процессорам (начиная с Intel 8086 в 1978 году) возникла необходимость в новой операционной системе. Исторически сложилось так, что ею стала не усовершенствованная CP/M, а совсем иная ОС, которую разработала фирма Microsoft, возглавляемая Биллом Гейтсом. И хотя в настоящее время MS-DOS практически вытеснена более новыми системами с графическим интерфейсом, ее поддержка по-прежнему сохраняется.

Необходимо подчеркнуть, что при написании MS-DOS программисты стремились достичь максимально возможной совместимости с предшествующей ей CP/M³. Нас сейчас данное свойство интересует в связи с тем, что в MS-DOS был написан функциональный аналог DDT — отладчик Debug. Таким образом, познакомившись с программой Debug, мы тем самым увидим, как выглядела работа с компьютером 2–3 десятилетия назад, когда никаких окон не было и в помине, а все действия инициировались не мышью, а набором текста в командной строке. Возможно, это не прибавит полезных навыков эксплуатации компьютера, но, согласитесь, весьма любопытно лишний раз почувствовать, как далеко ушел прогресс в интерфейсе общения человека и машины.

2. Особенности программы Debug

Debug представляет собой исполняемый файл, который запускается операционной системой стандартным образом. Тем не менее, будучи запущенным, он имеет некоторые особенности, которых нет у обычной прикладной программы. Обсудим их, воспользовавшись схемами, которые представлены на *рис. 1*.

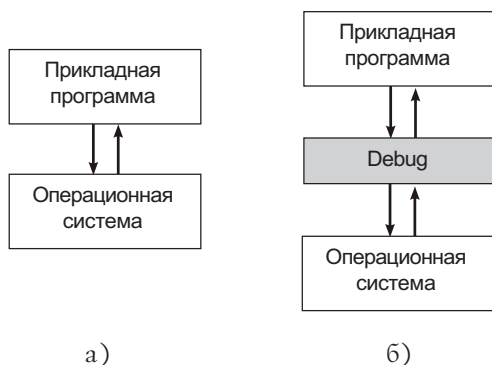


Рис. 1. Особенности функционирования программы Debug

² ОС CP/M написана для процессора Intel 8080; Z80 был более совершенным процессором, причем система его команд включала все инструкции процессора Intel 8080 в качестве подмножества.

³ Маленький пример: в MS-DOS, кроме штатной команды удаления `del`, имеется абсолютно эквивалентная ей команда `era` — именно такое имя использовалось в CP/M.

На *рис. 1а* показана ситуация с обычной прикладной программой. Пользователь дает операционной системе команду запустить программу⁴ и ОС запускает ее. Важно подчеркнуть, что система сохраняет контроль над прохождением приложения (иначе, в частности, вам не удалось бы закрыть его окно!). При своем функционировании программа активно опирается на сервисы операционной системы: обменивается через ее посредничество с внешними устройствами, запрашивает необходимую дополнительную оперативную память и т.д. Когда приложение завершает работу, оно сообщает об этом операционной системе и последняя выгружает его из памяти, освобождая все использовавшиеся ресурсы компьютера.

Для программы Debug ситуация несколько сложнее. Дело в том, что отладчик по своему назначению должен работать с программой пользователя, которая либо набирается с клавиатуры, либо читается из файла. В любом случае теперь мы имеем дело уже не с двумя, а с тремя программами (*рис. 1б*), причем Debug должен полностью контролировать отлаживаемую им программу. Контроль этот весьма разносторонний; здесь приведем только простой пример. Пусть Debug запустил программу пользователя, та нормально проработала и, завершаясь, собирается стандартным образом сообщить об этом операционной системе. Debug должен зафиксировать эту ситуацию, завершить работу отлаживаемой программы, перехватив управление и не давая операционной системе завершить сеанс работы отладчика. Аналогичные действия он должен проделать и в той ситуации, если в процессе работы программы пользователя возникла ошибка, иначе после любой самой пустяковой ошибки вам придется заново перезапускать Debug (особенно печальные последствия это бы имело в случаях, когда исправления в отлаживаемой программе не сохранялись на диске).

Таким образом, главная особенность программы Debug по сравнению с остальными приложениями состоит в том, что отладчик полностью⁵ контролирует работу другой (отлаживаемой) программы.

3. Что умеет Debug

Конкретные функции программы-отладчика подробно перечислены в статье [1], а также в пункте 1 при описании отладчика DDT (напомним, что функции DDT и Debug практически одинаковы).

⁴ Обычно запуск инициируется двойным щелчком мыши, но можно использовать и командную строку (как и в MS-DOS).

⁵ Полностью, конечно, только в идеале; реально вполне возможна ситуация, когда вследствие грубых ошибок в неотлаженной программе нормальное функционирование сеанса отладки нарушится.

Для дальнейшего обсуждения выделим наиболее важные группы действий отладчика:

- вывод на экран содержимого памяти в требуемом виде;
- изменение содержимого памяти;
- вывод на экран содержимого регистров микропроцессора;
- изменение содержимого регистров микропроцессора;
- запуск отлаживаемой программы;
- возврат управления отладчику после завершения программы пользователя или после прохождения определенной ее части;
- прочие действия (перевод мнемоник ассемблера⁶ в коды, копирование содержимого одной области памяти в другую, шестнадцатеричный калькулятор, работа с файлами и др.).

Рассмотрим подробнее, как реализует Debug некоторые из перечисленных действий.

3.1. Вывод и изменение содержимого памяти

Как следует из пункта 2, Debug обязан выделить определенный объем памяти под отлаживаемую программу. Здесь, казалось бы, нет проблем, поскольку любая программа вправе запросить у операционной системы необходимую память, и она ее обязательно получит. Трудность, однако, состоит в том, что данная память по смыслу имеет особый статус — это память прикладной программы. В связи с этим немедленно возникает целый ряд вопросов. Должен ли отладчик отображать и особенно изменять **только** эту память прикладной программы (скажем, разрешить ли пользователю работать с областями памяти, в которых находится сам отладчик или даже операционная система)? Как поступить, если работающая программа пользователя “злонамеренно” или по ошибке захочет изменить “чужой” участок памяти (допустим, “подправить” саму программу Debug)?

Ответ на все эти вопросы такой. Тот, кто пытается работать с программой на языке процессора, предполагается достаточно опытным пользователем, поэтому Debug не содержит “защиты от дурака” и не накладывает особых ограничений на выбор адресов памяти. Стоит, правда, добавить, что в многозадачной среде Windows, в отличие от MS-DOS, для которой изначально предназначался Debug, контроль памяти более жесткий. Так что чрезмерно грубые попытки вторжения в “чужую” область памяти будут пресечены на уровне ОС.

Из сказанного выше следует важное правило. Работа с программой под отладчиком Debug в случае

грубых ошибок может приводить к непредсказуемым ситуациям, поэтому **вы должны быть очень внимательны на всех этапах — от написания программы до ее набора с клавиатуры**. В случае большого ввода с клавиатуры дополнительно рекомендуем сохранить программу и данные на диске, прежде чем предпринимать дальнейшие манипуляции с отладчиком. К счастью, для наших несложных экспериментов последняя мера вряд ли потребуется.

3.2. Вывод и изменение содержимого регистров микропроцессора

Несмотря на отдельные трудности, распределение памяти между программой Debug и отлаживаемой программой все-таки можно произвести. С регистрами микропроцессора ситуация принципиально иная — их крайне мало, и они постоянно необходимы работающей программе. Так что не существует иной возможности сохранять значения регистров после остановки пользовательской программы, кроме как копировать эти значения в оперативную память (см. рис. 2). Отсюда первое, что делает отладчик при остановке работы программы пользователя, — это немедленно копирует содержимое всех регистров микропроцессора в память. Когда потребуются продолжить выполнение отлаживаемой программы, Debug заботливо восстановит предусмотрительно сохраненные значения и только потом запустит программу.

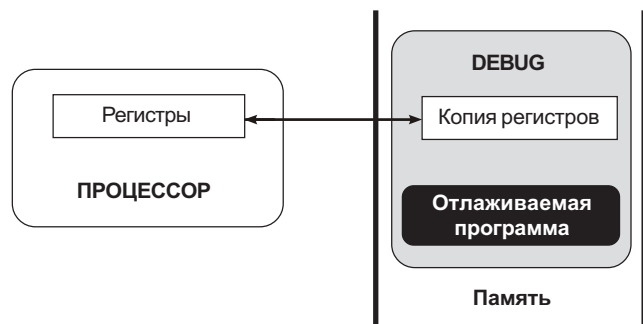


Рис. 2

Заметим, что Debug фактически создает в памяти своеобразную виртуальную машину, которая для исполнения программы использует центральный процессор.

3.3. Запуск отлаживаемой программы

Данная операция очень проста — достаточно выполнить переход на требуемый адрес и немедленно начнется выполнение нужной части программы (не забывайте о мерах, изложенных в пункте 3.2).

3.4. Возврат управления отладчику

Насколько проста предыдущая операция, настолько сложна данная! В самом деле, программа-отлад-

⁶ Термин “мнемоника” использован потому, что запись команды на языке ассемблера говорит о смысле этой команды. Например, запись **mov ah, 11** дает команду процессору занести число 11 в регистр **ah** (англ. *move* — перевозить, передвигать). — Ред.

чик запустила абсолютно неизвестную программу, и та начала работать. Как заставить ее автоматически вернуть управление отладчику?

При запуске отлаживаемой программы существуют три принципиально разных ситуации:

1) программа запускается без всякого контроля со стороны отладчика;

2) программа запускается с возможностью приостановки выполнения в заданных точках;

3) в программе пошагово выполняется заданное количество команд.

Интересно, что для каждой из них возврат из исполняемой пользовательской программы в отладчик происходит своим собственным путем. Рассмотрим эти пути подробнее.

Если пользователь уверен в том, что в его программе нет серьезных ошибок, грозящих непредсказуемыми последствиями, он может запустить программу на полное выполнение. Согласно принятым в MS-DOS соглашениям, программы завершаются специальной инструкцией INT 20h, которую отладчик и перехватывает. Таким способом после завершения программы отладчик снова получает управление и сеанс отладки продолжается.

Примечание. В некоторых источниках рекомендуется завершать программу командой RET (*return* — по-английски *возврат*). По мнению автора, это менее надежный вариант.

Второй сценарий отладки заключается в том, что программа запускается на исполнение, но в отличие от первого случая в ней предварительно устанавливаются так называемые *контрольные точки* (или точки останова — по-английски *breakpoint*). Если выполнение доходит до одной из них, то программа прерывается, и управление вновь переходит к отладчику. Описанным способом происходит выполнение определенной части программы (несколь-

ко точек приостановки требуется при ветвлении программы).

При реализации контрольных точек перед запуском отладчик запоминает данные из байтов, находящихся по их адресам, а затем заносит туда код специальной инструкции INT 3, которая обеспечивает выход в Debug. После получения управления отладчик удаляет из программы все инструкции прерывания, восстанавливая первоначальное содержимое контрольных байтов.

Наконец, третий, пошаговый режим запуска реализуется уже не программным, а аппаратным путем. В регистре флагов (см. [1]) имеется для этого специальный управляющий бит TF. “TF (*Trap Flag*) — флаг трассировки (пошагового режима). При его установке после выполнения каждой команды вызывается внутреннее прерывание типа 1 (INT 1)” [3]. В классических книгах по ассемблеру прямо указывается, что “Программа DOS DEBUG устанавливает данный флаг так, что возможно пошаговое выполнение каждой команды для проверки изменения содержимого регистров и памяти” [4].

Литература

1. Заславская О.Ю. Работа с регистрами центрального процессора. / “В мир информатики” № 48 (“Информатика” № 1/2005).
2. Уэйт М., Ангермейер Дж. Операционная система CP/M. М.: Радио и связь, 1986.
3. Гук М. Процессоры Intel: от 8086 до Pentium II. СПб.: Питер, 1997.
4. Абель П. Язык Ассемблера для IBM PC и программирования. М.: Высшая школа, 1992.

Продолжение — в следующем выпуске

Модель колеса. Часть 2 Моделирование

Л.Н. Медведев,
Москва

Напомню, что в предыдущей статье [1] на основе модели, рассматривавшей колесо как геометрическое тело, мы исследовали траектории движения точек колеса и увидели, что кривые получаются довольно интересные и сложные.

В этой статье мы посмотрим, что же происходит с колесом как с материальным телом, т.е. какие скорости имеют точки колеса и какие ускорения они получают во время движения. Зная законы Ньютона (надеюсь, что их знают и читатели!), в частности, второй, мы сможем определить силы, действующие на вещество, из которого изготовлено колесо.

Может быть, некоторым из вас покажется удивительным, как колесо продолжает существовать при таких силах...

Хотел бы обратить внимание, что в данный момент мы начинаем второй проход по этапам моделирования, перечисленным в учебниках [2, 3]. С учетом того, что рассказано в первой статье, начинаем сразу с раздела “Формализация модели”⁷, точнее — с корректировки разработанной ранее модели.

В данном случае колесо представляет собой то же тело, но материальное. Будем рассматривать “судьбу” маленькой части колеса, имеющей единичную массу и расположенной в точке А (см. рис. 1 на с. 41). Колесо катится вправо без проскальзывания.

⁷ См. этап 5 в первой статье. — Ред.

Скорость качения (т.е. та скорость, с которой перемещается центр колеса) равна V_0 . Радиус колеса R_0 . Зададим систему координат. Пусть ось абсцисс (x) будет находиться на высоте, на которой находится центр колеса, а ось ординат (y) направлена вверх и находится там, где в начале отсчета времени находится центр колеса. Точка A расположена от центра колеса на расстоянии r . Движение этой точки разобьем на два — вращение вокруг оси и перемещение вместе с осью колеса. Скорость вращения колеса ω связывает скорость качения колеса и его радиус.

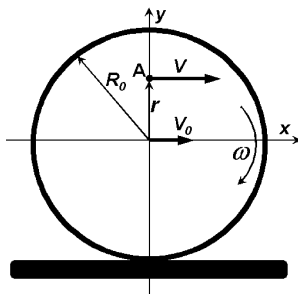


Рис. 1

Сначала вычислим скорость, с которой движется этот объем. Непосредственно скорость определить можно из уравнений для координат точки колеса, которые мы получили в первой статье.

$$\begin{aligned} x &= \omega R t + r \sin(\omega t) \\ y &= r \sin(\omega t) \end{aligned} \quad (1)$$

Составляющие скорости по горизонтали и по вертикали определяются производными соответствующих координат по времени. Пусть вас не пугает такое страшное выражение, все математические выкладки мы пропустим.

Итак,

$$\begin{aligned} V_x &= \omega R + r \omega \cos(\omega t) \\ V_y &= -r \omega \sin(\omega t) \end{aligned} \quad (2)$$

Соответственно, ускорения можно получить тоже взятием производных от составляющих скорости:

$$\begin{aligned} a_x &= -r \omega^2 \sin(\omega t) \\ a_y &= -r \omega^2 \cos(\omega t) \end{aligned} \quad (3)$$

Теперь надо построить диаграмму соответствующих уравнений. В качестве оси абсцисс выберем координату x , а по оси ординат будем откладывать значения V_x, V_y, a_x, a_y .

На рис. 2 приведена верхняя часть таблицы, по которой определяются соответствующие значения:

| | A | B | C | D | E | F |
|---|-----|---------|---------|---------|---------|---------|
| 1 | r= | 1 | 10 | | | |
| 2 | R= | 1 | 10 | | | |
| 3 | w= | 1 | 10 | | | |
| 4 | Пи= | 3,14159 | | | | |
| 5 | t | x | Vx | Vy | Ax | Ay |
| 6 | 0 | 0 | 12,5664 | 0 | 0 | -39,478 |
| 7 | 0,1 | 1,2161 | 11,3664 | -3,6932 | -23,205 | -31,939 |

Рис. 2

Для того чтобы было удобнее оформлять формулы в ячейках B6:F6 и они были понятнее, дадим имена ячейкам с параметрами модели (B1:B4). Это делается так: устанавливаем указатель активной ячейки на ячейку B1 и в окне адреса вводим имя ячейки (r_точки) так, как на рис. 3.

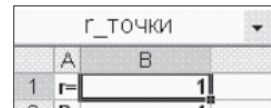


Рис. 3

Соответственно, ячейке B3 дадим имя Rколеса, B3 — частота, B4 — Пи. Откуда взялось π ? Дело в том, что для определения скорости точки надо использовать размерные величины, а угловая скорость вращения ω связана с частотой вращения F соотношением:

$$\omega = 2\pi F, \quad (4)$$

— поэтому наши формулы надо скорректировать, заменив везде ω на $2\pi F$. Новые формулы выглядят так:

$$V_x = 2\pi F R + r 2\pi F \cos(2\pi F t) \quad (2a)$$

$$V_y = -r 2\pi F \sin(2\pi F t)$$

$$a_x = -r (2\pi F)^2 \sin(2\pi F t) \quad (3a)$$

$$a_y = -r (2\pi F)^2 \cos(2\pi F t)$$

В ячейки B6:F6 введем формулы:

- B6: = 2*Pi*частота*Rколеса*A6+r_точки*SIN(2*Pi*частота*A6)
- C6: = частота*Rколеса*2*Pi+r_точки*частота*2*Pi*COS(2*Pi*частота*A6)
- D6: = -2*Pi*r_точки*частота*SIN(2*Pi*частота*A6)
- E6: = -4*Pi*Pi*r_точки*частота*частота*SIN(2*Pi*частота*A6)
- F6: = -4*Pi*Pi*r_точки*частота*частота* COS(2*Pi*частота*A6)

В ячейку A6 введем значение 0, в A7 — 0,1 и “протянем” этот ряд до величины 6,3.

Чтобы заполнить формулами нужный диапазон, выделим ячейки B6:F6 и дважды щелкнем по маркеру заполнения. Таблица готова.

Надо построить диаграммы скоростей и ускорений. Чтобы получить диаграмму скоростей, выделим диапазон B5:D69 и с помощью Мастера диаграмм построим графики скоростей:

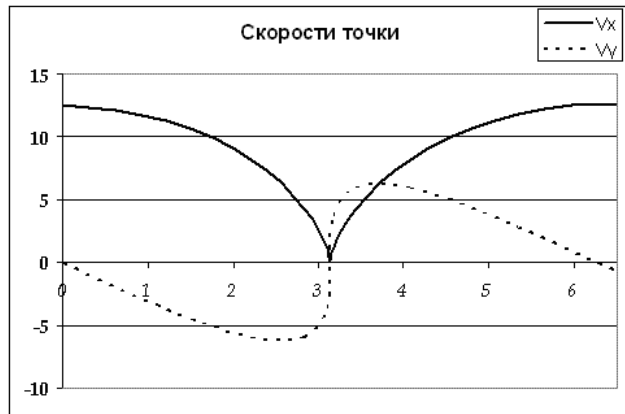


Рис. 4

А для построения графика ускорений надо выделить интервал со значениями x (B5:B69) и, держа нажатой клавишу **Ctrl**, выделить диапазон E5:F69, где содержатся данные для ускорений. Затем вызвать **Мастера диаграмм** и обычным путем построить график:

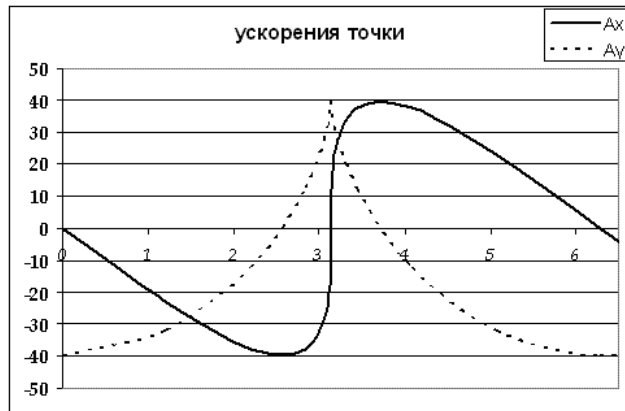


Рис. 5

Как сделать регуляторы, показанные на рис. 2, рассказано в предыдущей статье.

Модель готова к использованию. Проанализируем полученные графики. Обратим внимание на особенность около $x = 3,1$. В этом месте горизонтальная скорость имеет острый угол, а ускорение, соответственно, резко меняется от -40 до $+40$. В каких единицах эти значения? Если использовать систему Си, то значения скорости измеряются в м/с, а ускорения в м/с², т.е. при скорости качения около ≈ 13 м/с (≈ 46 км/ч) ускорение меняется от $-4g$ до $+4g$. Ничего себе рывок! Что же будет при увеличении скорости вдвое? Результаты — на рис. 6 и 7.

Ясно, что скорость удвоилась, а ускорение — учетверилось. Оно ведь пропорционально квадрату скорости. Рывок стал еще заметнее: от $-15g$ до $+15g$. Это уже что-то космическое. Например, космонав-

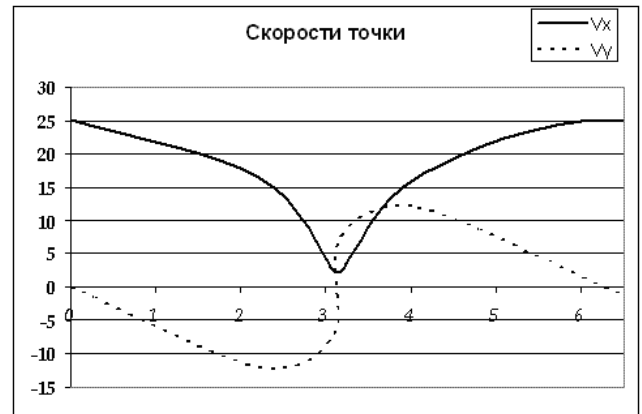


Рис. 6

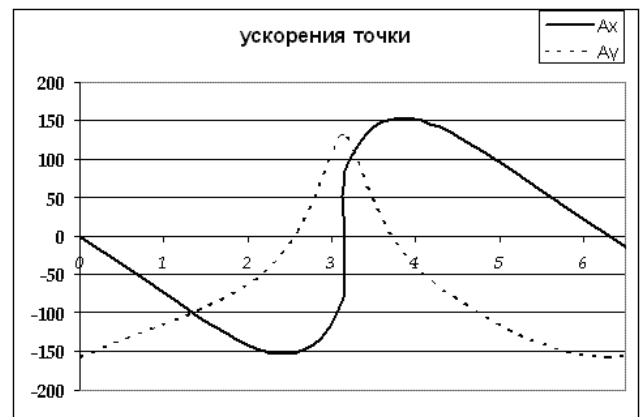


Рис. 7

ты при посадке испытывают ускорения менее $12g$. Что же произойдет при увеличении скорости еще вдвое? Для экономии газетного места приведу здесь только результаты: скорость 180 км/ч дает рывки от -600 до $+600$ м/с², т.е. от $-60g$ до $+60g$. Такие скорости — рядовое явление для болидов “Формулы-1”. Что же творится с колесом? Оно должно сжиматься и растягиваться (ведь оно резиновое, хотя и с металлическим каркасом). В результате в процессе движения колесо перестает быть круглым, а становится яйцевидным, вытянутым в направлении движения машины. Предлагаю вам выразить восхищение перед мастерством конструкторов и производителей, создавших такое колесо!

Литература

1. Медведев Л.Н. Модель колеса. / “В мир информатики” № 66 (“Информатика” № 1/2005).
2. Бешенков С.А., Кузьмина Н.В., Ракитина Е.А. Информатика. Систематический курс. Учебник для 11-го класса гуманитарного профиля. М.: БИНОМ. Лаборатория Знаний, 2002.
3. Бешенков С.А., Ракитина Е.А. Моделирование и формализация. Методическое пособие. М.: БИНОМ. Лаборатория Знаний, 2002.

*Ответы,
решения,
разъяснения*

**к заданиям,
опубликованным в газете
“В мир информатики”
№ 74, 75, 76
 (“Информатика”
№ 9, 10, 17/2006)**

1. Задача “Жестокий закон”

Крестьянин ответил так: “Я иду, чтобы быть пошешенным”. Если он сказал правду, то часовой должен его утопить, но в этом случае получится, что крестьянин сказал неправду, потому что его утопят, а не повесят. Если же ответ крестьянина является неправдой, то часовой должен повесить крестьянина, но тогда получится, что крестьянин сказал правду, т.к. его повесят.

Правильный ответ представили:

- Баженов Михаил, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;
- Деминцев Борис, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;
- Морозова Дина, Республика Татарстан, г. Казань, школа № 22, учитель **Осипова А.А.**;
- Шестакова Екатерина, г. Сибай, Республика Башкортостан, лицей “ПолиТЭК”, учитель **Кусябаева Г.С.**

2. Статья “Четыре приема быстрого счета”

2.1. Доказать с помощью формул справедливость приведенного в статье правила определения квадрата чисел, оканчивающихся на 5, в том числе применительно к трехзначным, четырехзначным, ... числам, можно следующим образом.

Представим двух-, трех- или четырехзначное число, оканчивающееся на 5, в виде $abc \cdot 10 + 5$, где abc — число, равное количеству десятков в представляемом числе (оно будет, соответственно, одно-, двух- и трехзначное). Тогда можем записать:

$$\begin{aligned}(abc \cdot 10 + 5)^2 &= (abc \cdot 10)^2 + \\ &+ 2 \cdot abc \cdot 10 \cdot 5 + 5^2 = (abc)^2 \cdot 100 + \\ &+ abc \cdot 100 + 5^2 = abc \cdot (abc + 1) \cdot 100 + 25.\end{aligned}$$

Итак, чтобы возвести рассматриваемое число в квадрат, надо число, образованное количеством десятков в нем, умножить на ближайшее большее целое число и к произведению приписать 25. Это правило распространяется и на аналогичные более чем четырехзначные числа.

Напомним, что оригинальный алгоритм возведения в квадрат трехзначных чисел, оканчивающихся на 5, предложила Евгения Кузнецова, ученица школы № 33 г. Ярославля, учитель **Головина Л.И.** (он был описан в

газете-вкладке “В мир информатики” № 79 / “Информатика” № 20/2006).

2.2. Правило умножения на 9 чисел в виде последовательности подряд идущих цифр от 1 до n ($2 \leq n \leq 9$) следующее: надо от числа, состоящего из $(n + 1)$ единиц, вычесть число, на 1 большее, чем n . Например:

$$\begin{aligned}1 \times 9 &= 11 - 2 \\ 12 \times 9 &= 111 - 3 \\ 123 \times 9 &= 1111 - 4 \\ 1234 \times 9 &= 11111 - 5\end{aligned}$$

...

Ответы прислали:

- Баженов Михаил, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;
- Морозова Дина, Республика Татарстан, г. Казань, школа № 22, учитель **Осипова А.А.**;
- Деминцев Борис, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**

3. Статья “Шестнадцать студентов”

Для большей наглядности удобно составить таблицу, в клетках которой надо проставить для каждого студента факультет, курс и любимый вид спорта:

| Город | Возраст | | | |
|----------|--------------------------|--------------------------------|------------------------------|---------------------------------|
| | 20 лет | 21 год | 22 года | 23 года |
| Тула | А. — — | Б. — — | В. биолог — — | Г. математик I шахматы |
| Орел | Д. — — — | Е. — — футбол | Ж. химик II шахматы | З. — — — |
| Курск | И. — — волейбол | К. геолог III шахматы | Л. — — — | М. — — — |
| Белгород | Н. — — — | О. — — — | П. — — — | Р. — — — |

В таблице приведены сведения, известные из условия задачи. Заполним пустые места (отмеченные черточкой).

Так как известны три шахматиста: курянин, орловец и туляк, а их возраст 21, 22 и 23 года, то четвертый шахматист из Белгорода и ему 20 лет. Значит, его фамилия — Н., он биолог (так как остальные три шахматиста: математик, химик, геолог) и учится на IV курсе (остальные шахматисты учатся на I, II, III курсах). Запишем эти сведения под номерами 1, 2 и 3 в таблицу:

| Город | Возраст | | | |
|----------|--|--|---|---|
| | 20 лет | 21 год | 22 года | 23 года |
| Тула | А. геолог (5) II (17) футбол (28) | Б. химик (4) IV (18) бокс (29) | В. биолог III (22) волейбол (30) | Г. математик I шахматы |
| Орел | Д. математик (6) III (16) бокс (27) | Е. биолог (13) I (19) футбол | Ж. химик II шахматы | З. геолог (12) IV (26) волейбол (31) |
| Курск | И. химик (7) I (15) волейбол | К. геолог III шахматы | Л. математик (8) IV (21) футбол (33) | М. биолог (9) II (25) бокс (34) |
| Белгород | Н. биолог (2) IV (3) шахматы (1) | О. математик (14) II (20) волейбол (32) | П. геолог (10) I (23) бокс (35) | Р. химик (11) III (24) футбол (36) |

Под номером 4 записываем, что Б. — химик (так как туляки — биолог и математик, а студенту К. 21 год и он геолог).

В этой ситуации легко определить специальности других студентов:

— так как студенты из Тулы имеют разную специальность, то А. — геолог (запишем это под номером 5);

— поскольку Д. не может быть химиком (из Орла химик — Ж.), то Д. — математик (6),

— так как среди 20-летних все имеют разную специальность, то И. — химик (7);

— поскольку Г. математиком быть не может, так как среди 23-летних студент такой специальности уже определен, то среди курян математик — Л. (8), а М. — биолог (9).

Рассуждая аналогично, получим, что П. — геолог (10), Р. — химик (11), З. — геолог (12), Е. — биолог (13), О. — математик (14).

После этого можно определить номер курса каждого студента. Химик из Курска И. — студент первого курса, так как курянин К. с III курса, химик Ж. со II курса, а двадцатилетний Н. с IV курса. Запишем это под номером 15. Аналогично определяют (под номерами 16–26) курсы, на которых учатся остальные студенты.

Осталось определить любимые виды спорта. Двадцатилетний студент из Орла Д. — боксер (27), так как орловцы Е. и Ж. — футболист и шахматист, соответственно, а двадцатилетний И. — волейболист. Далее в таблице указаны под номерами 28–36 виды спорта, которыми увлекаются студенты (например, 29: Б. — боксер, так как туляки А. и Г. — футболист и шахматист, а химик И. — волейболист и т.д.).

Правильные ответы прислали:

— Абдуллин Рамиль, средняя школа деревни Старый Бабиц Кармаскалинского р-на Республики Башкортостан, учитель **Абдуллин Р.Ф.**;

— Александров Алексей, г. Новочеркасск, Суворовское училище МВД РФ, преподаватель **Воронкова О.Б.**;

— Басакина Людмила, Боярскова Светлана, Докучаев Ростислав, Кудина Александра, Лисина Любовь, Нуязина Мария, Поповичева Александра, Старостенко Анна, Тихонов Роман, Черненко Иван и Шишкина Дарья, г. Волгоград, лицей № 10, учитель **Широкова Л.В.**;

— Бобу Андрей и Ищук Роман, Республика Татарстан, г. Елабуга, школа № 2, учитель **Титовская А.О.**;

— Бурцев Анатолий, Воронов Сергей и Ковалев Кирилл, г. Ярославль,

школа № 33, учитель **Ярцева О.В.**;

— Галанова Алла и Татаринов Иван, г. Ярославль, школа № 33, учитель **Цикина Е.Н.**;

— Гераськин Андрей и Маковкин Владимир, г. Новокузнецк Кемеровской обл., гимназия № 44, учитель **Дубовицкая Н.В.**;

— Глижинский Дмитрий, г. Бендеры, Республика Молдова, гимназия № 2, учитель **Глижинская С.Л.**;

— Гушин Дмитрий, Чкаловская средняя школа Асекеевского р-на Оренбургской обл., учитель **Диниева Г.Г.**;

— Дмитриев Юрий, средняя школа села Бима Агрызского р-на, Республика Татарстан, учитель **Дмитриева Э.А.**;

— Дубовицкий Николай и Пальчиков Алексей, г. Новокузнецк Кемеровской обл., гимназия № 44, учитель **Митина Н.В.**;

— Желнов Роман, г. Стерлитамак, Республика Башкортостан, Стерлитамакский художественно-технологический техникум, преподаватель **Орлова Е.В.**;

— Загайнова Наталья, Несмачных Татьяна и Савинова Мария, г. Белово Кемеровской обл., поселок Краснобродский, школа № 31, учитель **Зайцева Л.А.**;

— Идиятов Владимир, г. Ревда Свердловской области, школа № 28, учитель **Кольцова Е.М.**;

— Ильин Геннадий, Москва, г. Зеленоград, школа № 1194, учитель **Кулаженова Н.В.**;

— Исханова Гузель, средняя школа села Новые Клешки, Республика Башкортостан, учитель **Гарипова Р.М.**;

— Коптяева Екатерина, Тюменская обл., Ханты-Мансийский АО — Югра, Октябрьский р-н, село Шеркалы, Шеркальская национальная средняя общеобразовательная школа, учитель **Ефименко А.Н.**;

— Кузнецова Евгения, г. Ярославль, школа № 33, учитель **Головина Л.И.**;

— Максиманко Виктория, Мимивалеева Наталья и Селина Ксения, средняя школа поселка Дормидонтовка Вяземского р-на Хабаровского края, учитель **Афиногенова Н.И.**;

— Малявкин Георгий, поселок Надвоицы, Республика Карелия, школа № 1, учитель **Богданова Л.М.**;

— Мельник Алексей, г. Лениногорск, Республика Татарстан, школа № 8, учитель **Кашапова Р.Х.**;

— Михайленко Ирина, поселок Лимбяха Новоруренгойского р-на Тюменской обл., школа № 1, учитель **Крылова Л.Н.**;

— Михайлова Людмила, г. Сегежа, Республика Карелия, школа № 5, учитель **Меньшиков В.В.**;

— Пантюхина Анна и Савинов Владислав, г. Струнино Владимирской обл., школа № 11, учитель **Волков Ю.П.**;

— Пахомов Николай, г. Великий Новгород, средняя школа “Комплекс «Гармония»”, учитель **Королева О.В.**;

— Сазонова Екатерина, г. Ростов-на-Дону, школа № 109, учитель **Суслина Т.И.**;

— Филонов Евгений, средняя школа поселка Тавричанка Альшеевского р-на, Республика Башкортостан, учитель **Корнеева М.В.**;

— Шамшин Владимир, средняя школа рабочего поселка Пинеровка, Саратовская обл., Балашовский р-н, учитель **Пичугин В.В.**;

— Ямкина Анна, г. Ульяновск, школа № 37, учитель **Ямкина Е.В.**

4. Статья “Игра Баше”

Напомним, что в статье было предложено исследовать вариант игры Баше, в котором тот игрок, который своим ходом вынужден забрать последний предмет, проигрывает, а также разработать компьютерную программу, моделирующую такую игру.

Ответы и программы прислали:

— Алейников Андрей, г. Волгоград, лицей № 10, учитель **Широкова Л.В.**;

— Баженов Михаил, средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**;

— Бурцев Анатолий, Воронов Сергей и Ковалев Кирилл, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

— Глижинский Дмитрий, г. Бендеры, Республика Молдова, гимназия № 2, учитель **Глижинская С.А.**;

— Деминцев Борис, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

— Филонов Евгений, средняя школа поселка Тавричанка Альшеевского р-на, Республика Башкортостан, учитель **Корнеева М.В.**

Чтобы найти выигрышную тактику игры, определим, какие ситуации являются проигрышными для игрока, делающего очередной ход. Такой ситуацией, в частности, является положение, когда на столе остался один предмет. Если число забираемых предме-

тов не превышает k , то второй такой ситуацией являются наличие на столе $1 + (k + 1)$ предметов. Действительно, сколько бы предметов при этом не взял делающий очередной ход, его соперник может свести ситуацию к одному оставшемуся предмету. Проанализировав другие проигрышные ситуации — на столе $1 + 2(k + 1)$, $1 + 3(k + 1)$, ... предметов, можно установить, что все они характеризуются следующим свойством: остаток от деления числа имеющихся предметов на $(k + 1)$ равен 1. Следовательно, выигрышная стратегия заключается в том, чтобы брать столько предметов, чтобы оставлять сопернику проигрышную ситуацию (количество предметов, для которого справедливо только что указанное свойство). Ясно, что если уже исходное количество предметов обладает этим свойством, то начинающий игру не сможет реализовать описанную выигрышную тактику, если его соперник также знает ее.

Установим теперь правило, определяющее, сколько предметов надо взять, чтобы оставить сопернику проигрышную ситуацию (когда можно это сделать). Обозначим имеющееся в какой-то момент игры количество предметов m и напомним, что брать можно от 1 до k предметов. Можно рассуждать так. Количество предметов в проигрышной ситуации равно ближайшему меньшему чем m числу, кратному $(k + 1)$, увеличенному на 1. Это количество можно определить следующим образом:

если m кратно $(k + 1)$
то
 $(m \operatorname{div} (k + 1)) * (k + 1) + 1$ предметов

иначе

$((m \operatorname{div} (k + 1)) - 1) * (k + 1) + 1$ предметов,
— где div — операция целочисленного деления.

Можно упростить это правило так: надо оставлять сопернику $(m - 1) \operatorname{div} (k + 1) * (k + 1) + 1$ предметов.

Перейдем к разработке программы, которая моделирует игру Баше между компьютером и человеком. В ней, кроме указанных величин m и k , используем следующие:

n — начальное количество предметов;

b — количество предметов, которое берет делающий очередной ход;

p — величина, принимающая случайное значение 1 и 2 и определяющая участника игры, делающего первый ход;

z — количество предметов, которое надо оставить, чтобы создать для соперника проигрышную ситуацию (см. чуть выше).

алг Игра_Баше

нач цел n, k, m, p, b, z

| Исходные данные

$n := 15 + \operatorname{rnd}(10)$

вывод нс, "Исходное количество предметов: ", n

$k := 2 + \operatorname{rnd}(4)$

вывод нс, "Можно брать от 1 до ", k

| Определяем, кто начинает игру

```

p := 1 + rnd(2)
если p = 1
  то
    вывод нс, "Начинает компьютер"
  иначе
    вывод нс, "Начинаете Вы"
все
m := n
нц |Повторение ходов
  если p = 1
    то |Начинает компьютер
      |Проверяем, можно ли применить
      |выигрышную тактику
      если mod(m, k + 1) <> 1
        то |Можно
          |Применяем ее
          |Определяем проигрышную ситуацию
          z := div(m - 1, k + 1) * (k + 1) + 1
          |Берем необходимое количество
          |предметов
          b := m - z
        иначе
          |Выигрышную стратегию применить нельзя
          |Берем предметы "наугад"
          b := 1 + rnd(k)
      все
      вывод нс, "Компьютер взял", b
    иначе
      нц
        вывод нс, "Сколько Вы берете?"
        ввод b
        если b <= 0 или b > k или b > m
          то
            вывод нс, "Неправильное значение"
          все
        кц при b > 0 и b <= k и b <= m
      все
    |Новое значение m
    m := m - b
    |Проверяем, не закончилась ли игра
    если m = 0

```

```

то
  |Игра закончилась
  |Определяем победителя
  если p = 2
    то
      вывод нс, "Последний предмет
      |взяли Вы, т.е. "
      вывод "выиграл компьютер!"
    иначе
      вывод нс, "Последний предмет
      |взял компьютер, "
      вывод "т.е. Вы выиграли –
      |поздравляем!"
  все
  иначе |Игра продолжается
    вывод нс, "Осталось предметов: ", m
    |Изменяем величину p
    | (следующий ход делает другой
    | участник игры)
    если p = 1
      то
        p := 2
      иначе
        p := 1
    все
  кц при m = 0
кон

```

Предлагаем читателям исследовать вариант игры Баше, в котором взявший своим ходом оставшиеся предметы выигрывает. Определите выигрышную тактику игры и разработайте программу, моделирующую игру компьютера и человека. Результаты присылайте в редакцию.

Найти число

Сумма цифр 20-значного числа N равна 10, сумма цифр числа $7N$ равна 70, сумма цифр числа $19N$ равна 19. Найти N .

Внимание!
Конкурс

Конкурс № 53 для учащихся

Чемпионат СНГ по решению числовых ребусов (☺). Тур 2

Решите, пожалуйста, числовой ребус:

$$\begin{array}{r}
 \begin{array}{cccc|cc}
 M & Y & X & A & X & A \\
 X & A & & & Y & X & A \\
 \hline
 K & X & & & & & \\
 A & P & & & & & \\
 \hline
 Y & X & A & & & & \\
 Y & X & A & & & & \\
 \hline
 0 & & & & & &
 \end{array}
 \end{array}$$

В нем цифры заменены буквами. Одинаковым буквам соответствуют одинаковые цифры, разным буквам — разные цифры.

Ответ отправьте в редакцию до 1 января 2007 г. по адресу: 121165, Москва, ул. Киевская, д. 24, "Первое

сентября", "Информатика" или по электронной почте: inf@1september.ru. Пожалуйста, четко укажите в ответе свои фамилию и имя, населенный пункт, номер и адрес школы, фамилию, имя и отчество учителя информатики.

Внимание! Редакция решила рассматривать в качестве задания одного из туров конкурса № 53 ребус, опубликованный в газете-вкладке "В мир информатики" № 78 ("Информатика" № 19/2006). Участникам конкурса, не представившим решение указанного ребуса, предлагаем сделать это.

Итоги конкурса № 49 для учащихся будут подведены в следующем выпуске нашей газеты (этот номер готовился в октябре, и мы ждем поступления всех ответов).



Фестиваль педагогических идей «Открытый урок»

ВСЕ МАТЕРИАЛЫ БУДУТ ОПУБЛИКОВАНЫ!

Дорогие коллеги! Издательский дом «Первое сентября» приглашает вас принять участие в фестивале педагогических идей «Открытый урок» 2006/07 учебного года.

САМЫЙ МАССОВЫЙ ОТКРЫТЫЙ ПЕДАГОГИЧЕСКИЙ ФОРУМ!

В 2006/07 учебном году фестиваль «Открытый урок» проводится в четвертый раз. В фестивале 2003/04 учебного года приняли участие более 2600 педагогов, в 2004/05 учебном году — 4700, в прошлом году — более 5000.

Фестиваль стал самым массовым и представительным открытым педагогическим форумом. Материалы всех участников публикуются. **Каждый** участник **получает** полный комплект итоговых материалов, включающий:

- **персональный диплом;**
- **сертификат**, подтверждающий факт публикации материалов;
- **книги** — сборники тезисов всех статей;
- **компакт-диски** (CD-ROM) с полнотекстовыми версиями всех материалов.

В специальном разделе представлена информация об учебных заведениях.

Со всеми материалами можно ознакомиться на сайте фестиваля <http://festival.1september.ru>



КАК ПРИНЯТЬ УЧАСТИЕ В ФЕСТИВАЛЕ?

Если **вы** — педагог и хотите представить на фестиваль статью (несколько статей), пожалуйста, заполните форму индивидуальной заявки. Если вы планируете представить материалы в соавторстве с коллегами, заполнить форму заявки обязательно должен каждый. Сам факт отправки заявки вас ни к чему не обязывает (в том числе, конечно, и к участию в фестивале). Нет необходимости заранее принимать решение о том, сколько статей вы хотите опубликовать, будут они выполнены индивидуально или в соавторстве. В ответ на заявку вам будет выслано подробное положение о фестивале, в котором детально описаны все варианты участия. Фактически на этапе подачи заявки необходимо выполнить лишь одно условие: **каждый, кто имеет намерение участвовать в фестивале, должен отправить заявку.**

Если **вы** — директор школы, детского сада, учреждения дополнительного образования и т.п. и хотите представить на фестиваль педагогический опыт вашего образовательного учреждения и опубликовать о нем информацию справочного характера, пожалуйста, заполните форму заявки от учреждения. Как и в случае индивидуального участия, такая заявка вас ни к чему не обязывает и является лишь сообщением о намерении. В ответ на заявку будет выслано положение о фестивале с подробными условиями участия.

Индивидуальная заявка. Заполняется печатными буквами

ФАМИЛИЯ: _____
ИМЯ: _____
ОТЧЕСТВО: _____
ИНДЕКС: _____ АДРЕС: _____

✂ КОНТАКТНЫЙ ТЕЛЕФОН С КОДОМ: _____

Участвовали ли вы в фестивалях «Открытый урок» прошлых лет?
Если да, укажите, пожалуйста, номера соответствующих дипломов.
2003/04: _____ 2004/05: _____ 2005/06: _____

Заявка от учреждения. Заполняется печатными буквами

НАИМЕНОВАНИЕ УЧРЕЖДЕНИЯ (полное, в соответствии с уставом): _____

ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО ДИРЕКТОРА: _____

ИНДЕКС: _____ АДРЕС: _____

✂ КОНТАКТНЫЙ ТЕЛЕФОН С КОДОМ: _____

Участие в фестивале платное. Подробная схема расчета стоимости участия содержится в положении, которое высылается в ответ на заявку. Ниже приведены лишь рамочные параметры.

Для индивидуальных участников при расчете стоимости используются два ключевых понятия: оргзнос за участие и стоимость публикации одной статьи. Оргзнос за участие составляет 190 рублей. После оплаты оргзноса участник получает возможность опубликовать любое количество статей — индивидуально или в соавторстве. В сумму оргзноса также включена стоимость одного комплекта итоговых материалов — книг и компакт-дисков, которые получают все участники. Стоимость публикации одной статьи составляет 390 рублей. Обращаем внимание, что сумма указана именно за статью и взимается единожды вне зависимости от количества соавторов. Каждый автор статьи получит персональный диплом и сертификат, подтверждающий факт публикации.

Для участников фестивалей прошлых лет ПРЕДУСМОТРЕНА **100%-я СКИДКА** НА ОПЛАТУ ОРГЗНОСА.

Стоимость участия в фестивале образовательного учреждения составляет 2000 рублей.

Заявки следует направлять до 15 декабря 2006 г. (дата фиксируется по почтовому штемпелю предприятия-отправителя) по адресу: ул. Киевская, д. 24, г. Москва, 121165, ИД «Первое сентября», оргкомитет фестиваля «Открытый урок». Также можно подать заявку на сайте фестиваля. Справки по тел.: (495) 249-52-53 или по e-mail: festival@1september.ru

"КАК ЭТО ДЕЛАЮ Я"

Методический конкурс для учителей информатики

ЗАДАНИЕ ОДИННАДЦАТОГО ТУРА

"ПРОЕКТЫ"

Информатика — самый "проектный" из школьных предметов. Уникальные возможности информационных технологий позволяют широко вести проектную работу как внутри самого предмета, так и далеко за его формальными рамками. В разного рода интегративных проектах информатика выступает в качестве достойного партнера практически для любого предмета.

Примеров тому можно приводить множество. Ученики под руководством учителей информатики и других предметов разрабатывают презентации, сайты, демонстрационные программы, электронные учебные пособия, виртуальные лаборатории. Отдельный интерес представляют чисто информатические проекты (они, конечно, преимущественно связаны с программированием).

Расширение проектной деятельности — магистральный путь развития школы. И здесь, как и в большинстве новаций, информатики снова оказываются "на передовой".

Пожалуйста, поделитесь с коллегами своими наработками и идеями. В проектной деятельности новизна особенно важна. Не слишком интересно (не только для учителя, но и для учеников) по много раз повторять пройденное (хотя, конечно, нам известны и примеры удачных проектов, которые реализуются из года в год и никому не надоедают).

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 (примеры проектов) можно оформлять в виде приложений. Объем приложений не ограничен.

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, одиннадцатый 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 одиннадцатого тура — до 31 января 2007 г. (для "бумажных" писем дата отправки фиксируется по штемпелю предприятия-отправителя).

Результаты этого тура будут опубликованы в № 6/2007. На сайте "Информатики" материалы будут размещаться по мере поступления.

**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. Сёмина,
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
гл. ред. — С.В. Волков,
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. Бузоева,
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. ПИ № 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 09.11.2006.
Номер подписан 09.11.2006.
Заказ № 615523
Цена свободная

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