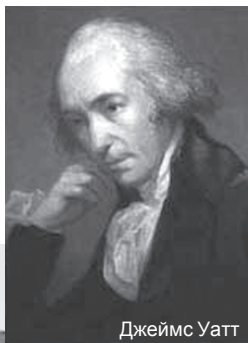
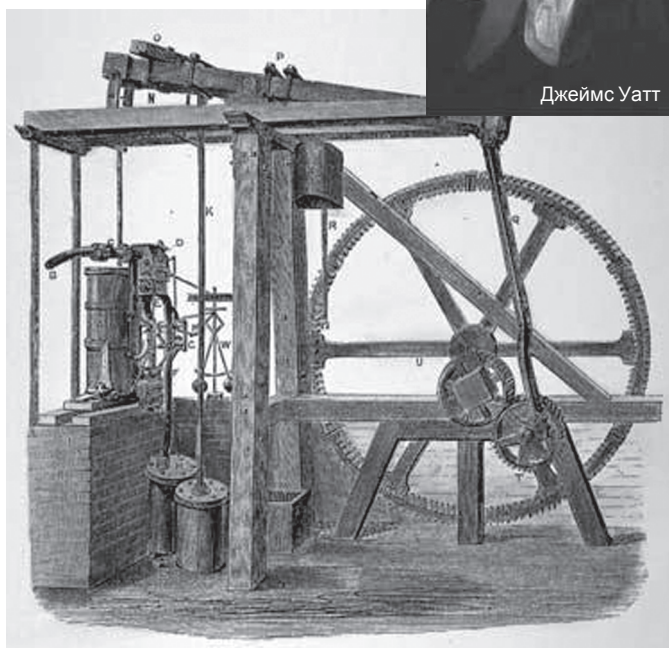


# Учителю информатики: памятные даты января

19 января 1736 года в шотландском городе Гриноке родился Джеймс Уатт (1736–1819) — выдающийся изобретатель (создатель универсального парового двигателя), чьим именем названа единица мощности в Международной системе единиц (*ватт*).



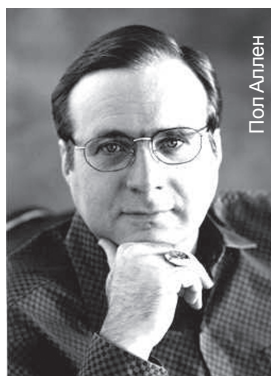
Джеймс Уатт



Паровая машина Джеймса Уатта,  
THE GRANGER COLLECTION, New York

21 января 1953 года в Сизтле, в семье библиотекаря родился Пол Аллен — один из основателей знаменитой Microsoft. До 30 лет биографии Пола Аллена и его школьного друга Билла Гейтса неразрывно связаны. Еще подростками они вошли в мир предпринимательства, написав программу для регулирования уличного движения и образовав компанию по ее распространению. Уже после школы (Пол Аллен учился тогда в Вашингтонском государственном университете) друзья создали первую операционную систему, разработав язык программирования BASIC для первого мини-компьютера — MITS Altair. В 1975 году произошло знаменательное событие — Аллен и Гейтс впервые использовали название Micro-Soft. В исходный код интерпретатора языка BASIC приятели включили и такую строку: "Micro-Soft BASIC: Bill Gates wrote a lot of stuff; Paul Allen wrote some other stuff". Позднее написание названия созданной ими компании изменилось на Microsoft, да и сфера интересов самой компании за прошедшее время тоже несколько расширилась.

В совместном бизнесе Пол Аллен занимался техническими идеями и перспективными разработками, Гейтсу ближе



Пол Аллен

Продолжение на с. 2

## СОДЕРЖАНИЕ НОМЕРА

### ПРЕДЛАГАЮ КОЛЛЕГАМ

**А.И. Сенокосов.** Слияние документов как лучшее средство от головной боли ..... 3–9  
**О.А. Житкова, Т.И. Панфилова.** VBA в приложении к Excel, Word и Power Point ..... 10–12

### ЭКСПЕРИМЕНТ

**В.Ф. Бурмакина, И.Н. Фалина.** Начало проекта по оценке ИКТ-компетентности девятиклассников ..... 13–16

### КОНКУРС

**“Как это делаю я”.**

**Методический конкурс для учителей информатики.**  
Задание V тура ..... 17  
Итоги II тура ..... 18

**Е.С. Леонова.** Общий план урока “Техника безопасности” ..... 19–20

**В.В. Трофимова.** Изучаем и повторяем правила техники безопасности ..... 20–21

**Л.З. Загвоздина.** Знакомство с правилами техники безопасности ..... 21

**Е.В. Ямкина.** Инструктаж по технике безопасности ... 21–22

**Г.А. Беркутова.** Соблюдаем технику безопасности ..... 22–23

**К.В. Чернобазова.** Соблюдаем технику безопасности ..... 23

### НА СТЕНД В КАБИНЕТЕ ИНФОРМАТИКИ

**К.В. Чернобазова.** Соблюдаем технику безопасности ..... 24

**Л.З. Загвоздина.** Знакомство с правилами техники безопасности ..... 25

### СЕМИНАР

**С.Б. Гашков.** Системы счисления и их применения ..... 26–30

### “НАЧАЛКА” № 2

*Газета-клуб для всех, кто учит информатике маленьких детей*

**А.В. Могилев.** “Мир информатики” для младших школьников ..... 31–36

### “В МИР ИНФОРМАТИКИ” № 67

*Газета для пытливых учеников и их талантливых учителей*

**Школа программирования Н.М. Тимофеева.** Как решать задачи? ..... 37–39

### Эксперименты

**Е.А. Еремин.** Работаем без операционной системы (!) ..... 40–43

**Задачник**  
Однофамильцы в поезде ..... 43

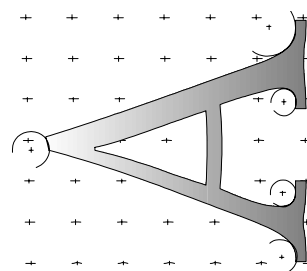
**“Ломаем” голову**  
Буквенный ребус ..... 44

**Внимание! Конкурс**  
Конкурс № 42 для учащихся ..... 44

Итоги конкурса № 40 для учащихся ..... 44–47

№ 2 (507)

16–31 января 2006



Методическая газета для учителей информатики

# ИНФОРМАТИК

## Учителю информатики: памятные даты января

Продолжение. См. с. 1

оказались переговоры, контракты и прочее деловое общение. Вероятно, тяга Аллена к науке в ущерб коммерции стала причиной того, что позже созданные им самостоятельно компании так и не смогли достичь того успеха, которого добилась Microsoft. И это при том, что компании Пола Аллена представляли и представляют на рынке программы, реализующие оригинальные и по-своему революционные идеи. Зато, по признанию самого Аллена, он всю жизнь придерживается напутствия, данного ему отцом: “Занимайся только тем, что тебе по-настоящему нравится”.

**23 января 1862 года** в городке Велау близ Кёнигсберга родился знаменитый немецкий математик **Давид Гильберт (1862–1943)** — “последний могианин” классической математики.

“Давид Гильберт занимает совершенно особое место в истории науки. Хотя большую часть жизни этот исследователь жил в XX веке, он продолжал оставаться “последним могианином” классической математики. Подобно своим великим предшественникам, К.Гауссу и Б.Риману, он имел универсальный склад мышления. В его жизни не оставалось места для увлечений — все свое время ученый отдавал математике” [3].

Исследования Гильберта оказали большое влияние на развитие целого ряда разделов математики, причем его деятельность в Гёттингенском университете способствовала тому, что Гёттинген в первой трети XX столетия являлся одним из основных мировых центров математической мысли [1]. Диссертации многих крупных математиков (в том числе Г.Вейля и Р.Куранта) были написаны под руководством Гильберта.

В 1900 году на Международном конгрессе математиков в Париже Гильберт поставил 23 проблемы [2, 4]. В своем докладе он старался отразить направления математических исследований предыдущих десятилетий и наметить пути дальнейших исследований. Сегодня некоторые из проблем Гильберта решены, другие все еще ждут решения.

В течение нескольких лет Гильберт сам предпринимал попытки решить ряд проблем вариационного исчисления и дифференциальных уравнений, а также теории интегральных уравнений. Построенная Гильбертом теория интегральных уравнений с симметричными ядрами стала одной из основ современного функционального анализа. Дополнительным результатом этой работы было формирование и введение в математику понятия, известного теперь как *пространство Гильберта*, или *гильбертово пространство*.

Последние годы своей жизни Гильберт посвятил проблеме, которую считал важнейшей для математики, а имен-

но созданию ее логических основ. В 1934 и 1939 годах вышли два тома “Оснований математики”, написанных Гильбертом совместно с П.Бернайсом. Вся дальнейшая работа, связанная с формированием логических основ математики, в значительной мере идет в направлении, наметенном Гильбертом, и использует предложенные им концепции.

**19 января 1912 года** в Петербурге родился **Леонид Витальевич Канторович (1912–1986)** — выдающийся математик и экономист, академик, лауреат Нобелевской премии, разработавший теорию так называемого “крупноблочного программирования”.

Данная теория предполагала оперирование не с отдельными числами и символами, а со структурами данных (матрицами, векторами, деревьями и т.д.), чьи элементы обрабатывались с помощью стандартных подпрограмм [5]. Заложенные в эту технологию идеи во многом предопределили развитие программирования на последующие годы (в частности, они используются в функциональном программировании).

**28 января 1857 года** в Рочестере (штат Нью-Йорк) родился **Уильям Бэрроуз (1857–1898)**, занимающий одно из главных мест в ряду изобретателей клавишных счетных машин.

Изобретение клавишного ввода чисел (взамен медленного ручного) для счетных машин позволило в середине 80-х годов XIX столетия наладить их промышленный выпуск. Свой вклад в создание клавишных машин внесли изобретатели многих стран, и видное место среди них занимает американец Уильям Бэрроуз. Машина Бэрроуза являлась двухтактной: на первом такте осуществлялась установка числа с помощью клавиш, а на втором — установленное число движением приводного рычага переносилось на счетчик. Таким образом, клавиши не имели отношения к действиям, выполняемым самой машиной, и оставались в опущенном положении с момента установки числа. Это позволяло непосредственно осуществлять контроль ввода и в случае необходимости исправлять ошибки. Машина могла печатать исходные числа, производить сложение (или вычитание) и печатать результат. Использовать ее для выполнения умножения было трудно.

Не раз предпринимались попытки усовершенствовать машину Бэрроуза. Удалось, например, расширить перечень

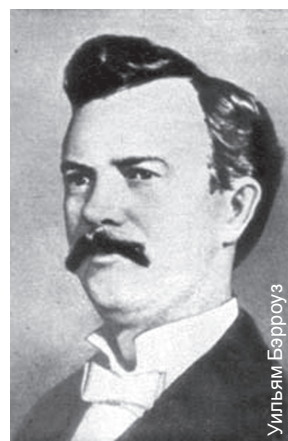


Давид Гильберт

*Hilbert*



Леонид Витальевич Канторович



Уильям Бэрроуз

# Слияние документов как лучшее средство от июньской головной боли

А.И. СЕНОКОСОВ,  
г. Екатеринбург

Как известно, Microsoft Office — это очень большой ящик с самыми разнообразными инструментами. И вряд ли найдется человек, который знал бы их все досконально (возможно, за исключением самих разработчиков). Оно и понятно: большинство пользователей счастливы даже оттого, что способны красиво оформить и распечатать простейший документ типа служебной записки или скромного делового письма.

Почти во всех школьных учебниках и программах традиционно считается, что многочисленные возможности MS Office — не предмет для изучения в базовом курсе информатики. И это совершенно справедливо. В противном случае мы рискуем подменить процесс обучения (и, предположительно, процесс развития мышления) скучной процедурой достаточно бездумного запоминания того, в какой последовательности необходимо нажимать кнопки и выбирать пункты меню.

Тем не менее вышеупомянутый богатейший инструментальный ящик вовсе не должен оставаться закрытым за семью замками. Вполне возможно, что хотя бы краткий его обзор может стать предметом профильного курса информатики и/или информационной технологии.

Вот и сегодняшняя публикация посвящена одному из слабостребованных средств MS Word, а именно — слиянию документов.

Разумеется, о слиянии документов при создании массовой рассылки корреспонденции наверняка слышали многие. Как правило, слышали, что в единственное письмо подставляется информация из базы данных, и это позволяет создать пухлую пачку корреспонденции с индивидуальными адресами и обращениями. Точно так же большинство уверено в том, что подобный инструмент просто необходим сомнительным фирмам, занимающимся торговлей по каталогам. Впрочем, уважаемые фирмы тоже скорее всего этим самым слиянием пользуются. А вот школам оно ну совершенно не нужно...

Думается, это несколько поспешное суждение, и для начала давайте рассмотрим несколько совершенно типичных ситуаций.

1. Нахлынула плотная волна районных олимпиад и научно-практических конференций, принесла вашим ученикам (а также и учителям) призовые места. Безусловно, эти достижения необходимо отра-

зить на общешкольном стенде информации в виде примерно такого объявления:

## Поздравляем с успешным выступлением на районной олимпиаде по информатике!

Ученики нашей школы заняли следующие места:

**Иванов Вячеслав (10-й “Б” класс) — 1-е место**  
(учитель П.Ф. Чудесный),

**Кузьмина Анастасия (9-й “А” класс) — 1-е место**  
(учитель Е.А. Чуткова),

**Чабанов Дмитрий (8-й “В” класс) — 2-е место**  
(учитель П.Ф. Чудесный).

**Молодцы!**

Как правило, подобный текст набирается один раз, а затем постоянно правится в зависимости от названия очередного мероприятия и фамилий призеров. А между тем эта информация вовсе даже не заслуживает того, чтобы о ней сразу же и забывали. Во-первых, она почти наверняка потребуется для школьного отчета. Во-вторых, очень может быть, что какое-нибудь 4-е или 5-е место районный оргкомитет не отметит даже грамотой, а для школы это достижение. И тогда ученика было бы неплохо поощрить хотя бы школьной грамотой на ближайшем празднике. В-третьих, итоги выступления учеников вполне достойны помещения в школьную историю.

Не будем пока предлагать свой вариант создания таких объявлений, а рассмотрим еще тройку ситуаций:

2. В школе прошел кросс “Золотая осень” (лыжные гонки “Жаркая зима”, соревнования “Серебряные коньки”, турнир “Ласковый кулак”, пожарное многоборье “Зажигай, чтобы не погасло”...). Естественно, что существуют списки с результатами участников. Еще более естественно наградить грамотами победителей соревнований в каждом классе. Классов много, призеров — еще больше, и печать грамот превращается в серьезную проблему для учителя физкультуры (и/или информатики).

3. Идет подготовка выпускного вечера. Для 9-го или, еще хуже, 11-го класса. Каждому ученику, кроме аттестата, очень хотелось бы вручить еще и скромную (или не очень) грамоту хотя бы за что-нибудь, чем он отличился в школе. Здесь же (в смысле, на выпускном вечере) принято поблагодарить (в письменном виде) и

подавляющее большинство родителей, которые активно помогали школе (например, практически добровольно сдавали деньги в фонд школы). Печать нескольких сотен грамот превращается в головную боль не только классных руководителей, но и администрации с учителями информатики в придачу.

4. В уважающих себя и, соответственно, учеников школах хорошим тоном стало вывешивать списки именинников. Как ни странно, самым простым способом автоматизировать это на редкость полезное дело является то же самое слияние документов.

Собственно говоря, третья ситуация несравнима с остальными, поэтому и название статьи посвящено именно ей. Как ни странно, технологически именно третья проблема решается особенно просто и эффективно. И по многим причинам имеет смысл начать именно с нее.

Итак, для пушей торжественности мероприятия на выпускные вечера закупается пара-другая сотен грамот примерно такого вида (см. рис. 1). Далее с помощью MS WORD печатается пробная страничка:



Рис. 1. Примерный образец грамоты

Несложная операция просмотра грамоты и странички на просвет позволяет подогнать буквы под рисунок грамоты, после чего компьютерно грамотный старшеклассник (а лучше — парочка ответственных старшеклассников), с трудом разбирая каракули классных руководителей и время от времени переверывая фамилии, начинают неблагодарный труд печати грамот, который частенько растягивается на несколько дней, а то и ночей.

Эта очень увлекательная и творческая работа вполне может стать значительно менее трудоемкой (за счет экономии бланков грамот).

Давайте для начала определим параметры, которые меняются от грамоты к грамоте.

1. Окончание обращения (либо *-к*, либо *-ца*).
2. Класс.
3. Имя и фамилия.
4. Повод для награждения.
5. Инициалы и фамилия классного руководителя.

Попросим от каждого класса список награждаемых, созданный в MS Excel (см. рис. 2).

Грамоты					
	A	B	C	D	E
1	Имя, фамилия	Окончание	Класс	Повод для награждения	Инициалы, фамилия классного
2	Василий Алексеев	к	11а	Выдающиеся достижения в перетягивании каната	Н.К.Панова
3	Семен Баранов	к	11а	Неоднократные победы в олимпиадах по перетягиванию каната	Н.К.Панова
4	Анна Ветлугина	ца	11а	Самоотвеженный труд по озеленению кл. территории	Н.К.Панова
5					
6					

Рис. 2. Документ MS Excel

Во избежание излишне творческого подхода настоятельно рекомендуется требовать заполнения уже готового электронного шаблона, который надо заранее подготовить. Заметим, что поля “Класс” и “Классный руководитель” просто дублируются протягиванием вниз.

Теперь возвращаемся к подготовленной грамоте и выбираем пункт меню “Сервис || Письма и рассылки || Слияние”. В правой части экрана появится окно “Слияние”. Нас будет интересовать список этапов, появившийся в нижней части этого окна.

1. Щелкаем мышкой по надписи “Далее. Открытие документа”.
2. Щелкаем мышкой по надписи “Далее. Выбор получателей”.
3. Чуть повыше щелкаем мышкой по надписи “Обзор”, расположенной под словами “Используйте имена и адреса из файла или базы данных”.
4. Выберите файл с заполненным шаблоном. Не забудьте, что в стандартном файле MS Excel есть три листа и необходимо выбрать правильный (см. рис. 3).

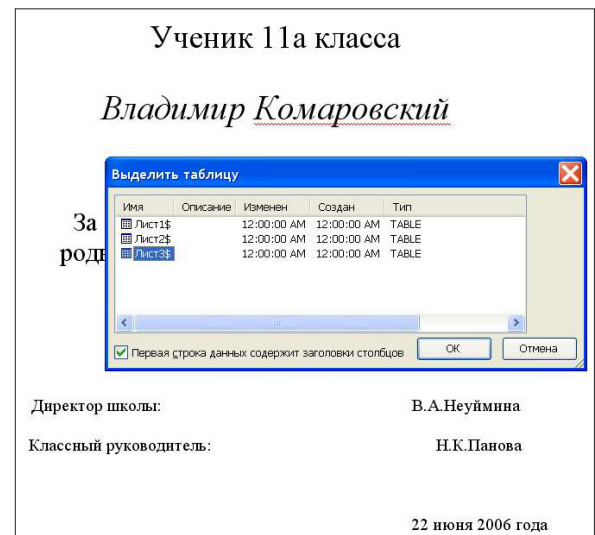


Рис. 3. Подключение файла с информацией для грамот

При подключении файла не забудьте указать, что нам требуются только непустые строки. Для этого в окне документа сначала щелкните мышкой по синему треугольнику возле любого поля, а затем выберите пункт “Непустые” (см. рис. 4).

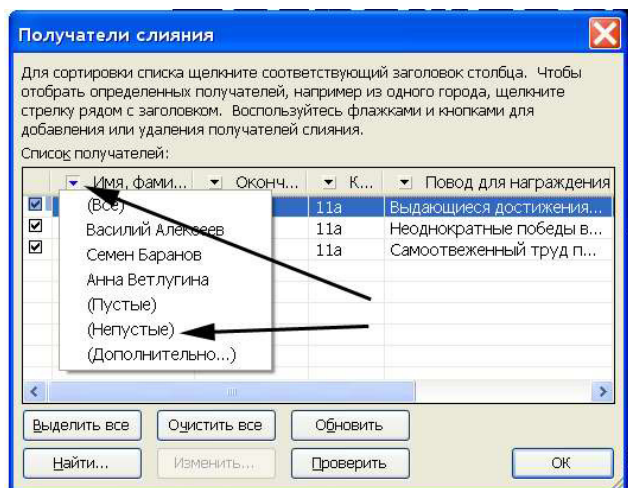


Рис. 4. Выбор только непустых полей

5. После подключения файла появится надпись: “В настоящее время получатели выбраны из...”. Теперь можно переходить к следующему этапу, щелкнув мышкой по надписи “Далее. Создание письма”.

6. Выделяем мышкой окончание -к и щелкаем по надписи “Другие элементы”. В появившемся окне выбираем “Окончание” и нажимаем кнопку “Вставить”.

7. Закроем окно и повторим аналогичную процедуру с каждым изменяемым полем.

Документ примет вид, подобный тому, что приведен на рис. 5.

Видно, что в процессе слияния часть полей почему-то стала располагаться не по центру. Исправьте это. Возможно, нарушилось и еще что-нибудь.

8. Щелкаем мышкой по надписи “Далее. Прочитать писем”. Возможно, потребуется вернуться назад для корректировки писем.

9. Щелкаем мышкой по надписи “Далее. Завершение слияния”.

10. Теперь имеет смысл не торопиться с печатью, а щелкнуть мышкой по надписи “Изменить часть писем”.

11. Проверьте все полученные грамоты. Измените при необходимости часть из них, особенно если повод для награждения оказался излишне длинным и не разместился в рамках грамоты. Заметим, что именно этот повод может нарушить формат грамоты, поэтому имеет смысл всегда делать его, скажем, трехстрочным, добавляя, если по-

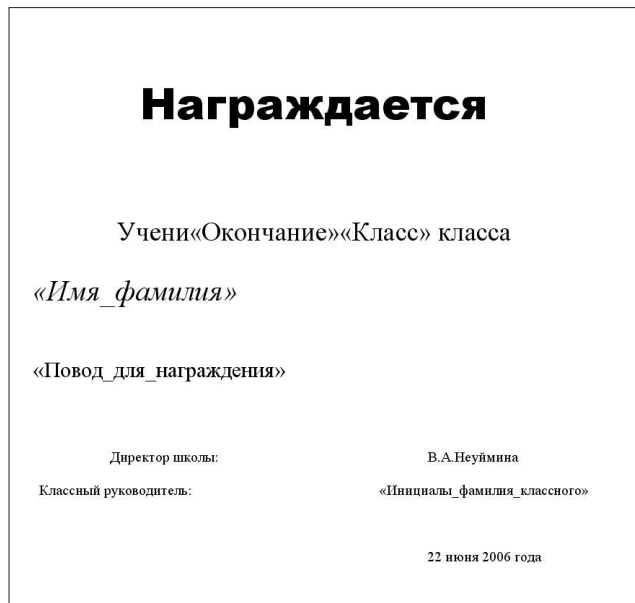


Рис. 5. Вид документа со вставленными полями

требуется, пустые строки. Проверка грамот, помимо всего прочего, позволяет осуществить известную в издательствах технологию “вторых глаз”, когда текст документа проверяется другим человеком, не тем, кто его написал.

12. Отправляйте документ на печать, не забывая подгружать в принтер пачки грамот.

В конце работы сохраните драгоценный шаблон, который сослужит вам еще добрую службу с грамотами такого типа. В дальнейшем можно просто подключать к документу другие файлы с аналогичными листами MS Excel и в считанные минуты напечатать огромное количество грамот.

Итак, с одной из самых жгучих проблем мы разобрались. Вернемся к лыжным гонкам. Итоговый протокол гонок на 3 километра может выглядеть, например, так:

Фамилия, имя	Время
Иванов Вася	13.20,3
Бобров Сева	14.01,5
Копылов Саша	13.24,3
Альбин Тема	12.50,5
Ахунов Саша	15.40,5
Бихмутов Артем	17.23,1
Бельтиков Дима	13.30,4
Боярцинов Артем	13.04,2
Валиев Тимур	14.23,3
Глухов Костя	11.55,2
Мефед Вова	12.04,9
Махеев Максим	11.37,2

Молчаливо предполагаем, что этот протокол подготовлен в MS Excel. Чтобы использовать данную таблицу для уже полюбившейся, надеемся, операции слияния, необходимо ее немного доработать.

1. Щелкнем мышкой по столбцу “Время” и, воспользовавшись кнопкой “Сортировка по возрастанию”, получим следующий вариант:

<i>Фамилия, имя</i>	<i>Время</i>
Махеев Максим	11.37,2
Глухов Костя	11.55,2
Мефед Вова	12.04,9
Алыбин Тема	12.50,5
Боярцинов Артем	13.04,2
Иванов Вася	13.20,3
Копылов Саша	13.24,3
Бельтиков Дима	13.30,4
Бобров Сева	14.01,5
Валиев Тимур	14.23,3
Ахунов Саша	15.40,5
Бихмутов Артем	17.23,1

2. Кроме того, в ячейку C2 поместим формулу =СТРОКА()-1.

3. Продолжим формулу вниз:

<i>Фамилия, имя</i>	<i>Время</i>	<i>Место</i>
Махеев Максим	11.37,2	1
Глухов Костя	11.55,2	2
Мефед Вова	12.04,9	3
Алыбин Тема	12.50,5	4
Боярцинов Артем	13.04,2	5
Иванов Вася	13.20,3	6
Копылов Саша	13.24,3	7
Бельтиков Дима	13.30,4	8
Бобров Сева	14.01,5	9
Валиев Тимур	14.23,3	10
Ахунов Саша	15.40,5	11
Бихмутов Артем	17.23,1	12

Эта дополнительная доработка занимает меньше минуты, но таблица становится гораздо более наглядной.

На базе этой новой таблицы создадим объявление с поздравлением и шаблон для печати грамот. Будем считать, что с шаблоном мы разобрались. А вот объявление создадим на базе другой технологии: вставки файла с обновлением. Итак, мы хотим получить примерно следующее:

**Поздравляем победителей и участников  
лыжных гонок “Жаркая зима”**  
(ст аршая группа)

<i>Фамилия, имя</i>	<i>Время</i>	<i>Место</i>
Махеев Максим	11.37,2	1
Глухов Костя	11.55,2	2
Мефед Вова	12.04,9	3
Алыбин Тема	12.50,5	4
Боярцинов Артем	13.04,2	5
Иванов Вася	13.20,3	6
Копылов Саша	13.24,3	7
Бельтиков Дима	13.30,4	8
Бобров Сева	14.01,5	9
Валиев Тимур	14.23,3	10
Ахунов Саша	15.40,5	11
Бихмутов Артем	17.23,1	12

4. Выберем нужный фрагмент таблицы в документе Excel. Скопируем его в буфер обмена.

5. Напечатаем поздравление, вставим буфер обмена в документ и в выпадающем меню параметров вставки (открывающемся в нижнем правом углу фрагмента вставки) выберем пункт “Использовать стиль конечной таблицы и связать с Excel”.

То, что получится после этой операции, нуждается в дальнейшей доработке:

**Поздравляем победителей и участников  
лыжных гонок “Жаркая зима”**  
(ст аршая группа)

<i>Фамилия, имя</i>	<i>Время</i>	<i>Место</i>
Махеев Максим	11.37,2	1
Глухов Костя	11.55,2	2
Мефед Вова	12.04,9	3
Алыбин Тема	12.50,5	4
...	...	...

6. Разместим таблицу по центру, уменьшим шрифт, выделим победителей и шрифт заголовков:

<b><i>Фамилия, имя</i></b>	<b><i>Время</i></b>	<b><i>Место</i></b>
<b>Махеев Максим</b>	<b>11.37,2</b>	<b>1</b>
<b>Глухов Костя</b>	<b>11.55,2</b>	<b>2</b>
<b>Мефед Вова</b>	<b>12.04,9</b>	<b>3</b>
Алыбин Тема	12.50,5	4
Боярцинов Артем	13.04,2	5
Иванов Вася	13.20,3	6
Копылов Саша	13.24,3	7
Бельтиков Дима	13.30,4	8
Бобров Сева	14.01,5	9
Валиев Тимур	14.23,3	10
Ахунов Саша	15.40,5	11
Бихмутов Артем	17.23,1	12

Мало того, что мы получили аккуратную таблицу в поздравлении, но за счет своей связи с файлом MS Excel поздравление сразу же меняется, как только меняются данные в исходной таблице. Давайте для примера изменим результат победителя с 11.37,2 на 12.57,2, не забыв после этого поставить курсор на вторую колонку и нажать кнопку “Сортировать по возрастанию”:

Фамилия, имя	Время	Место
Глухов Костя	11.55,2	1
Мефед Вова	12.04,9	2
Алыбин Тема	12.50,5	3
Махеев Максим	12.57,2	4
Боярцинов Артем	13.04,2	5
Иванов Вася	13.20,3	6
Копылов Саша	13.24,3	7
Бельтиков Дима	13.30,4	8
Бобров Сева	14.01,5	9
Валиев Тимур	14.23,3	10
Ахунов Саша	15.40,5	11
Бихмутов Артем	17.23,1	12

Как видите, в файле MS Word сохранилось исходное форматирование, а выделенными оказались уже новые победители.

На базе подготовленной таблицы Excel легко сделать и шаблон для печати грамот. Надеемся, это не вызовет у вас никаких трудностей.

Теперь перейдем к олимпиадам. Возможно, самый простой способ аккуратно учитывать все выступления своих учеников — это создать отдельный файл в формате MS Excel примерно такого формата (см. рис. 6):

	А	В	С
1	Районная по математике	Класс	Место
2	Петров Алексей	7	2
3	Семенова Даша	8	3
4	Блинд Стас	9	3
5	Юрченко Вадим	9	1
6	Ивашкевич Евгений	10	4
7	Шишелякин Евгений	10	2
8	Левин Дмитрий	11	3
9	Васильева Катя	11	1
10	Варшавский Денис	11	

Рис. 6. Учет выступлений на олимпиадах

Как видите, на отдельных листах этого файла предполагается собрать результаты выступлений вообще на всех олимпиадах всех уровней в текущем учебном году.

Перед тем как создать поздравительный плакат, произведем некоторую доработку таблицы. Конкретнее, отсортируем строки сразу и по классу, и по месту, занятому на олимпиаде. Выберем пункт меню “Данные || Сорти-

ровка” и в выпадающем окне “Сортировка диапазона” укажем сортировку по классу, а затем — по занятому месту (см. рис. 7).

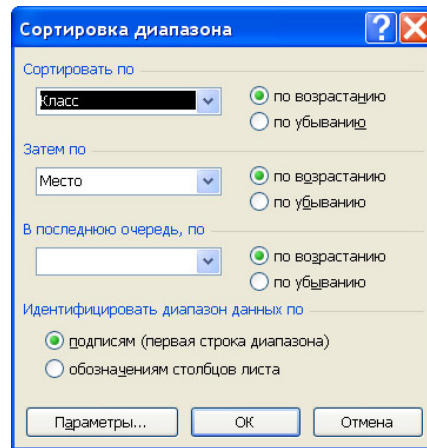


Рис. 7. Окно “Сортировка диапазона”

Добавим еще и колонку “Учитель”, а если необходимо — “Класс” и “Окончание”, как мы это делали в процессе подготовки грамот к выпускному вечеру (см. рис. 8).

Получим лист, пригодный и для подготовки поздравительного плаката, и для печати грамот. Разумеется, придется делать поздравления и грамоты отдельно для каждого предмета. Причем, для того чтобы не делать эту работу вторично, файл Excel с таблицей и соответствующие ему файлы MS Word с плакатами поздравлений и грамот, сделанных на его основе, имеет смысл хранить в одной папке. А в следующем учебном году скопировать все в другую папку и изменить данные, приведя их в соответствие с олимпиадами уже нового учебного года, автоматически получив весь пакет нужных документов. Заметим, что при формировании грамот мы вполне можем выбрать только те строки таблицы, которые нам необходимы. Это позволяет не дублировать грамоты, если они уже вручены районным или городским оргкомитетом соответствующей олимпиады.

И, наконец, рассмотрим, каким образом можно с помощью слияния организовать поздравления именинников.

	А	В	С	Д
1	Фамилия, имя	Класс	Место	Учитель
2	Петров Алексей	7	2	В.П.Чуткий
3	Семенова Даша	8	3	Н.Д.Семенова
4	Юрченко Вадим	9	1	В.П.Чуткий
5	Блинд Стас	9	3	В.П.Чуткий
6	Шишелякин Евгений	10	2	Н.Д.Семенова
7	Ивашкевич Евгений	10	4	Н.Д.Семенова
8	Васильева Катя	11	1	Е.К.Власова
9	Левин Дмитрий	11	3	Е.К.Власова
10	Варшавский Денис	11	3	Е.К.Власова
11				

Рис. 8. Возможный окончательный вид таблицы

В качестве пещки, от которой мы начнем очередной танец, возьмем следующую таблицу:

Фамилия	Имя	Дата рождения	Класс
Щенников	Эдуард	20.11.89	10-й "А"
Чабанов	Дмитрий	01.12.90	9-й "В"
Овсянников	Никита	03.01.91	8-й "Б"
Яковлев	Александр	17.11.88	11-й "А"
Ожиганов	Артем	27.11.90	9-й "А"
Козлов	Павел	22.11.98	1-й "А"

Разумеется, в настоящей таблице, как это ни страшно звучит, должны быть все ученики школы. Опыт, правда, показывает, что при наличии журналов это работа для одного старшего класса на один урок.

Задача состоит в том, чтобы:

- определить, у кого из учеников в ближайшем будущем будет день рождения;
- посчитать, сколько же лет исполняется имениннику;
- создать файл в формате Word, содержащий список именинников.

Мы объединим первую и вторую задачи и создадим колонку, в которой будет отображаться количество лет, которое "стукает" только тому школьнику, чей день рождения отстоит от сегодняшней даты не более чем, скажем, на неделю.

Разумеется, сегодняшнюю дату в формате ДД.ММ.ГГ мы можем получить, используя функцию СЕГОДНЯ() из MS Excel. Переведем и ее, и дату рождения в числовой вид с помощью функции ЗНАЧЕН(), мы даже сможем вычесть одно из другого. Получится при этом, правда, вовсе не то, что мы ожидаем. А именно — сколько полных дней прошло от дня рождения до сегодняшнего дня.

Стало быть, для достижения указанной цели (посчитать, сколько дней прошло от сегодняшнего дня до дня, когда ПРАЗДНОВАЛСЯ или будет ПРАЗДНОВАТЬСЯ день рождения) мы должны разработать нечто вроде собственной метрики. Для этого умножим месяц дня рождения на 30 и добавим к нему день месяца. То же самое сделаем и с сегодняшней датой. Вот модуль разности этих двух чисел и имеет смысл сравнивать, скажем, с числом 7. Не надо только забывать, что вышеуказанная методика не сработает при опасной близости к концу года, поэтому последний день, когда ее можно применять, 24 декабря. Аналогично первым днем является 7 января.

Итак, еще раз, но уже с использованием формул. Мы хотим знать, верно ли вот такое выражение:

```

|месяц_дня_рождения*30 +
число_дня_рождения – текущий_месяц*30 –
сегодняшнее_число| < 7

```

Если да, то в соответствующей колонке мы проставим разницу:

Текущий\_год – год\_дня\_рождения.

Если же нет — ничего проставлять не будем. Вернее, проставим пустую символьную константу.

Необходимое нам выражение с использованием функций Excel довольно громоздко, но, надеемся, не вызовет особых трудностей в понимании:

```

=ЕСЛИ(ABS(МЕСЯЦ(ЗНАЧЕН(СЗ))*30+
ДЕНЬ(ЗНАЧЕН(СЗ))- МЕСЯЦ(ЗНАЧЕН(СЕГОДНЯ()))*
30-ДЕНЬ(ЗНАЧЕН(СЕГОДНЯ())))<7;
ГОД(ЗНАЧЕН(СЕГОДНЯ()))-ГОД(ЗНАЧЕН(СЗ));"")

```

Скопировав эту изящную формулу во все строки (что делается одним легким движением мыши), мы получим следующую таблицу:

Фамилия	Имя	Дата рождения	Класс	Лет
Щенников	Эдуард	20.11.89	10-й "А"	16
Чабанов	Дмитрий	01.12.90	9-й "В"	15
Овсянников	Никита	03.01.91	8-й "Б"	
Яковлев	Александр	17.11.88	11-й "А"	
Ожиганов	Артем	27.11.90	9-й "А"	15
Козлов	Павел	22.11.98	1-й "А"	7

Можно сказать, с первой частью работы мы успешно справились. Хотя, если подумать, придуманная нами метрика весьма своеобразна. Так, если наша "поздравлялка" будет запущена 25 января, то, вопреки ожиданию, она поздравит и тех, кто родился 1 февраля. Впрочем, это даже хорошо. Немного хуже то, что в невисокосный год 24 февраля она не поздравит тех, кто родился 1 и 2 марта, хотя должна бы..

Не слишком усложняя себе жизнь, просто изменим число 30 на 28 и обеспечим выполнение народной мудрости "Лучше перепоздравить, чем недопоздравить".

А теперь переходим к созданию файла с поздравлением. Итак, хотелось бы видеть много строк подобного содержания:

**Поздравляем с 17-летием!**  
**Яковлев Александр 11-й "А" 17.11**

Реализуем слияние в точно таком же варианте, что и при подготовке грамот на выпускной вечер:

1. Печатаем текст: "Поздравляем с ".
2. Выбираем "Сервис || Письма и рассылки || Слияние" и подключаем таблицу с днями рождения, не забыв указать, что надо выбирать только строки с непустыми значениями в столбце "Лет".
3. В процессе работы получим вот такой шаблон:

```

Поздравляем с "Лет"-летием!!!
"Фамилия" "Имя" "Дата_рождения"

```

Просматривая уже готовый документ, неожиданно выясним, что "Дата\_рождения" почему-то подставляется в закеанском формате:



**Поздравляем с 16-летием!!!  
Эдуард Щенников 11/20/1989**

Возможно, где-то в недрах MS Word и существует возможность легким движением мышки исправить формат на европейский, но нам проще добавить еще две колонки в таблицу: “Месяц” с формулой =МЕСЯЦ(С3) и “Число” с формулой =ДЕНЬ(С3). Тогда исходная таблица примет окончательный вид:

Фамилия	Имя	Дата рождения	Класс	Лет	Число	Месяц
Щенников	Эдуард	20.11.89	10-й “А”	16	20	11
Чабанов	Дмитрий	01.12.90	9-й “В”	15	1	12
Овсянников	Никита	03.01.91	8-й “Б”		3	1
Яковлев	Александр	17.11.88	11-й “А”		17	11
Ожиганов	Артем	27.11.90	9-й “А”	15	27	11
Козлов	Павел	22.11.98	1-й “А”	7	22	11

4. На основе уже этой таблицы создадим шаблон:

**Поздравляем с “Лет”-летием!!!  
“Фамилия” “Имя” “Число” “Месяц”**

Как вы помните, слияние обеспечит печать вышеприведенных строк на отдельном листе. Нам это вовсе

перед созданием списка имен для слияния необходимо определить, какая из программ Office больше всего подходит для решения этой задачи. Для слияния можно использовать список, созданный в Word, Microsoft Access, Outlook или Microsoft Excel. Только что мы узнали, как пользоваться средством слияния документов, когда источником данных является файл, созданный в Microsoft Excel. Однако, если список не слишком велик и не нужны сложные условия сортировки, смело можно пользоваться операцией слияния внутри редактора Word.

Документ, содержащий текст и рисунки, которые должны быть одинаковыми в каждой версии конечного документа, называется основным документом слияния. Находясь внутри него, зайдём в меню “Сервис | Слияние” и в графе “Основной документ” обратимся к пункту “Создать”. Далее (для печати грамот, например) выберем “Документы на бланке” и “Активное окно”, поскольку мы уже находимся в заготовленном шаблоне. Теперь необходимо создать новый источник данных для слияния: таблицу, содержащую изменяющиеся параметры нашего документа (список учащихся). Для этого нажмём кнопку “Получить данные” и выберем команду “Создать источник данных”. В источник будут введены все перечисленные поля данных. Просмотрите имеющиеся в шаблоне: вы можете пользоваться ими, удалив ненужные поля и изменив их

даже и ни к чему. Поэтому, как и раньше, выберем пункт меню “Изменить часть писем” и в “Правке” заменим “Разрыв раздела” (надо нажать кнопку “Специальный” в окне “Найти и заменить”) на пустой символ.

Нажимаем “Заменить все” и получаем следующий документ:

**Поздравляем с 16-летием!!!  
Эдуард Щенников 20.11**

**Поздравляем с 15-летием!!!  
Дмитрий Чабанов 1.12**

**Поздравляем с 15-летием!!!  
Артем Ожиганов 27.11**

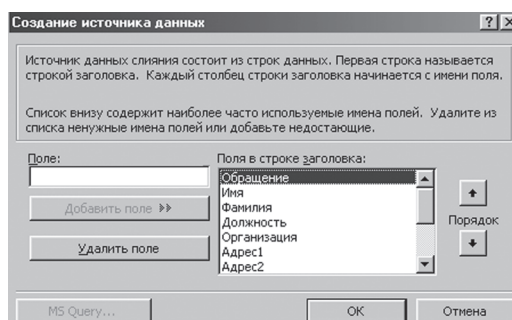
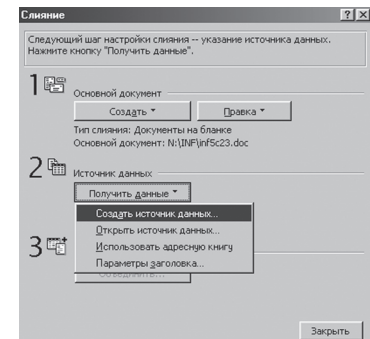
**Поздравляем с 7-летием!!!  
Павел Козлов 22.11**

Разумеется, при желании сюда можно добавить ещё и класс. Единственное неудобство — приходится постоянно открывать исходную таблицу и указывать, что выбираются лишь строки с непустым значением в колонке “Лет”.

Надеемся, наш короткий рассказ о слиянии убедил вас в полезности этого инструмента и, возможно, он в дальнейшем даже немного облегчит вашу жизнь.

порядок в списке (с помощью кнопок со стрелкой), или создать свои. Собственно, определение полей — основная задача при создании источника данных, так что нет необходимости весь список набирать в предложенной форме; достаточно нескольких записей. Результат сохранится в привычной таблице формата Word, с которой можно продолжать работать в редакторе, пользуясь всеми его благами (копированием полей, добавлением и удалением строк и т.д.).

Далее Word потребует добавить поля слияния в основной документ. Для этого опять используйте панель инструментов “Слияние”: необходимо просто отметить позиции, соответствующие источнику данных.



Думается, что процедура слияния внутри Microsoft Word не требует излишне подробных описаний: уже после двух-трех попыток документ будет иметь желаемый вид, а полученный результат вполне компенсирует затраченные усилия.

И ещё один совет: не забывайте о такой простой и полезной функции пакета Microsoft Office, как “Справка”!

# VBA в приложении к Excel, Word и Power Point

О.А. ЖИТКОВА, Т.И. ПАНФИЛОВА,  
Москва

Продолжение. Начало в № 1

## Занятие № 2. Создание макросов и их применение в Excel

На этом занятии мы рассмотрим следующие вопросы:

- Понятие макроса.
- Запись простых макросов.
- Выполнение макросов.
- Редактирование макросов.
- Назначение макроса графическим изображениям.
- Удаление макросов из списка макросов.
- Ограниченность макросов.

### Понятие макроса

Прежде чем приступить к написанию программ на VBA, воспользуемся простой возможностью создания программы (макроса) на языке VBA с использованием MacroRecorder.

MacroRecorder — это стандартное средство записи макросов в Excel, посредством которого можно записывать последовательность действий пользователя и получать соответствующий код (программу) на VBA. Программа, созданная с помощью MacroRecorder, называется макросом.

Основное назначение макросов — автоматизация работы пользователя. Кроме этого, созданный код макроса может служить основой для дальнейших разработок.

При записи макроса запоминаются все действия пользователя, будь то нажатие клавиши или выбор определенной команды меню, которые автоматически преобразуются в программный код на языке VBA.

Каждому макросу дается имя, а для быстрого запуска макроса можно создать или присвоить ему “горячую” клавишу (клавишу, по нажатию на которую будет производиться запуск макроса). После запуска макрос будет автоматически выполнен тем приложением, в котором он создан и запущен. При выполнении макроса компьютер воспроизведет все действия пользователя.

**Макрос** — это именованная последовательность заданных пользователем команд и действий, хранящаяся в форме программы на языке VBA.

### Запись макросов в приложении Excel

Для работы с макросами в приложении Excel имеется специальная панель.

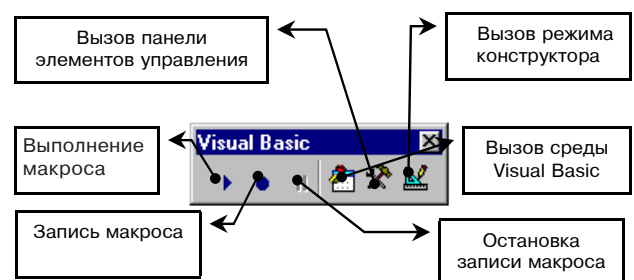


Рис. 1. Панель для работы с макросами

Для вызова этой панели необходимо выполнить последовательно команды меню: “Вид | Панель инструментов | Visual Basic”.

Для записи макроса необходимо:

1. Нажать на кнопку “Запись макроса” на панели Visual Basic.
2. В диалоговом окне “Запись макроса” присвоить имя макросу. (В имени макроса первым символом должна быть буква, не допускается использование пробелов.)
3. Выполнить действия, которые нужно записать.
4. Нажать кнопку “Остановить запись”.

### Пример

Рассмотрим процедуру записи последовательности действий на следующем примере: определим максимальное значение в выделенном диапазоне ячеек.

В таблице на с. 11 представлена информация о самых ценных алмазах в мире. Нам требуется создать макрос для нахождения алмаза с максимальной массой в граммах (рис. 2).

Это действие в Excel можно выполнить с помощью стандартной функции МАКС, но мы продемонстрируем на этом примере, как сохранить последовательность действий пользователя и на их основе создать макрос.

Выполните следующие действия:

1. Откройте новую книгу.
2. Создайте таблицу по приведенному образцу.
3. В ячейке D10 наберите “Максимальная масса в граммах”. Курсор после набора текста может находиться в любой ячейке, кроме ячейки, в которой мы хотим получить результат. На панели Visual Basic нажмите на кнопку “Запись макроса” (рис. 2).

	A	B	C	D	E
1	Название	Страна происхождения	Когда найден	Масса в каратах	Масса в граммах
2	Великий Могол	Индия	XVII век	787,0	157,40
3	Джонкер	Южная Африка	1934	726,0	145,20
4	Звезда Сьерра-Леоне	Западная Африка	1972	968,9	193,78
5	Куллинан	Южная Африка	1905	3106,0	621,20
6	Президент Варгас	Бразилия	1938	726,6	145,32
7	Река Уойе	Западная Африка	1945	770,0	154,00
8	Эксельсиор	Южная Африка	1893	971,5	194,30

Появится диалоговое окно “Запись макроса”:

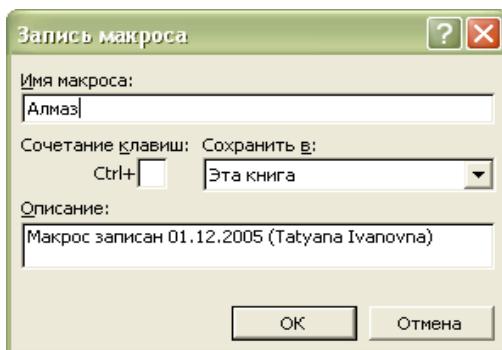


Рис. 2. Окно “Запись макроса”

Введите вместо названия “Макрос1” название макроса “Алмаз” и нажмите “ОК”. На экране появится панель инструментов “Остановить запись”. Теперь все производимые нами действия будут записываться до тех пор, пока не будет нажата эта кнопка. **Не выполняйте никаких лишних действий. Все ваши действия фиксируются и записываются.**

Установите курсор в ячейку E10. В этой ячейке должен появиться результат.

4. Выполните команду “Вставка | Функция”. Появится диалоговое окно “Мастер функций”. В окне “Категории функций” выберите “Статистические”. Затем выберите функцию МАКС. Появится диалоговое окно, в первой строке которого необходимо указать диапазон, в котором мы будем искать максимальное значение, — E2:E9 (см. рис. 3). Нажмите “ОК”.

5. Щелкните на кнопке “Остановить запись” на панели Visual Basic. Запись макроса завершена.

### Выполнение макросов

При выполнении макроса Excel повторяет те же действия, которые пользователь выполнял в процессе его создания. Продолжим работу с созданным макросом.

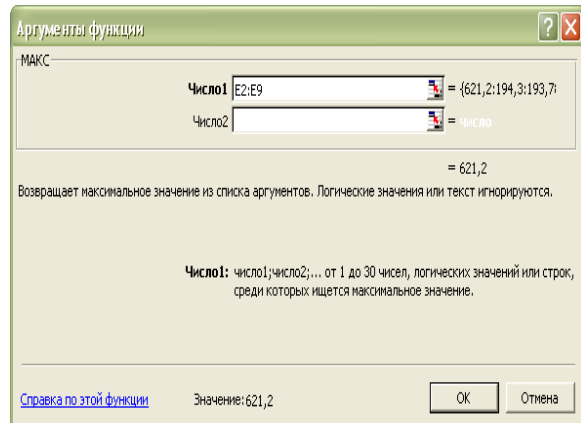


Рис. 3. Диалоговое окно функции МАКС

Чтобы проверить, как работает созданный нами макрос, прежде всего надо удалить из ячейки E10 полученный результат.

Затем:

1. Установите курсор в любую ячейку листа.
2. Выполните команду “Сервис | Макрос | Макросы”. Появится диалоговое окно, показанное на рис. 4:

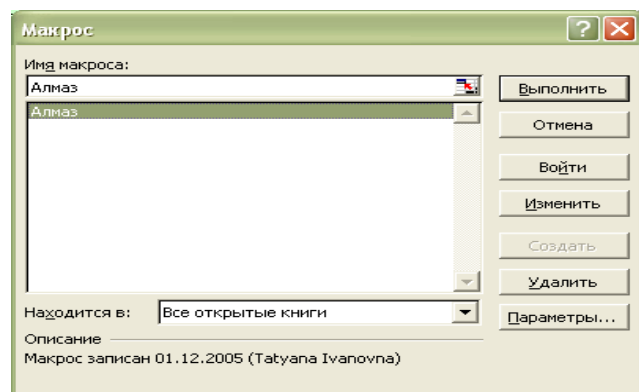


Рис. 4. Диалоговое окно “Макрос”

3. Выделите макрос “Алмаз” и щелкните по кнопке “Выполнить”. В ячейке E10 появится результат.

### Редактирование макросов

Во время записи макроса Excel запоминает ваши действия и преобразует их в код VBA. Можно просмотреть полученный код и отредактировать его, если в этом есть необходимость. Для просмотра созданного макроса выполните команду “Сервис | Макрос | Макросы”. Появится уже знакомое диалоговое окно “Макрос”.

Выделим макрос “Алмаз” и щелкнем по кнопке “Изменить”. Откроется окно редактора (см. рис. 5 на с. 12).

При записи макроса выполнялись всего два действия. Сначала мы установили курсор в ячейку E10. На языке VBA этому действию соответствует строка: `Range("E10").Select`

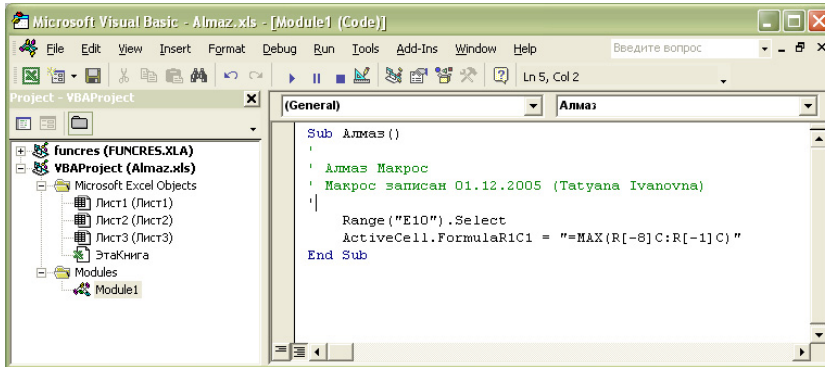


Рис. 5. Использование редактора Visual Basic для просмотра и редактирования кода VBA

Затем мы вызвали функцию МАКС и в качестве аргумента указали диапазон ячеек E2:E9:

```
ActiveCell.FormulaR1C1 = "=MAX(R[-8]C:R[-1]C)"
```

Полученный код можно редактировать непосредственно в редакторе Visual Basic. Изменим размер шрифта в ячейке E10 на 16:

```
Range("E10").Font.Size = 16
```

Можно также изменить цвет шрифта. Например, изменим цвет текста в ячейке на красный:

```
Range("E10").Font.ColorIndex = 3.
```

Значения, которые может принять свойство Font.ColorIndex, изменяются от 1 до 56.

После внесенных изменений текст макроса стал следующим:

```
Sub Алмаз ()
' Алмаз Макрос
Range("E10").Select
ActiveCell.FormulaR1C1 =
    "=MAX(R[-8]C:R[-1]C)"
Range("E10").Font.Size = 16
Range("E10").Font.ColorIndex = 3
End Sub
```

Закройте окно редактора, вернитесь на лист Excel, удалите содержимое ячейки E10 и запустите макрос на выполнение.

### Назначение макроса графическим изображениям

Разрабатывая приложение, надо думать о том, чтобы создать легкий и удобный интерфейс для выполнения задач автоматизации. Запуск макросов на выполнение посредством команд меню или кнопок на панели Visual Basic — не слишком удобный механизм для пользователей.

Для запуска макроса можно использовать любой элемент, находящийся на рабочем листе. Например, можно использовать для запуска любое графическое изображение. Рассмотрим соответствующий пример.

Используя панель “Рисование”, рисуем на листе любую автофигуру. Щелкнем правой кнопкой мыши на изображении и в открывшемся контекстном меню выберем команду “Назначить макрос”. Отобразится диалоговое окно “Назначить макрос объекту”. Затем необходимо выбрать макрос “Алмаз” и щелкнуть вне графического изображения, чтобы снять выделение с объекта.

Теперь макрос “Алмаз” можно запустить посредством щелчка на автофигуре (см. рис. 6).

Запустить макрос на выполнение можно и другим способом, используя командную кнопку. Вставить командную кнопку непосредственно на рабочий лист Excel можно с помощью панели инструментов “Формы”. Затем необходимо назначить ей наш макрос.

### Удаление макросов из списка макросов

Для того чтобы удалить макрос, надо:

1. Выполнить команду “Сервис | Макрос | Макросы”. Появится уже знакомое диалоговое окно.
2. Выделить макрос, подлежащий удалению, и щелкнуть по кнопке “Удалить”.
3. Подтвердить выполнение операции в специальном окне, которое появится.

### Ограниченность макросов

Множество операций Excel можно автоматизировать посредством макросов. Но макросы имеют и ограничения. С помощью макросов нельзя выполнить различные действия в зависимости от содержимого ячейки, нельзя вызвать некоторые диалоговые окна Excel, например, сохранить документ, нельзя отобразить и использовать пользовательские формы ввода данных.

Эти ограничения приводят к необходимости создания программ на VBA.

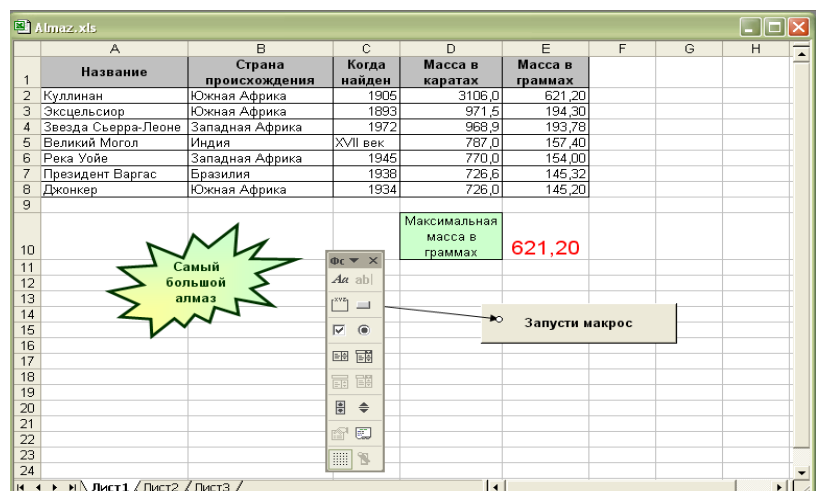


Рис. 6. Автофигура и кнопка для запуска макроса

# Начало проекта по оценке ИКТ-компетентности девятиклассников

В.Ф. Бурмакина, И.Н. Фалина,  
Москва

В рамках проекта “Информатизация системы образования” (ИСО), проводимого правительством Российской Федерации, с ноября 2005 года начало разрабатываться новое направление — “Разработка инструмента для оценки компетентности школьников в области ИКТ”.

В техническом задании к новому проекту указывается, что “...одним из индикаторов успешности проекта ИСО является его влияние на формирование информационной и коммуникационной компетентности выпускников девятого класса, которые завершают обучение по программе основной школы. Этот индикатор позволяет косвенно оценить влияние изменений в содержании и методах учебной работы в основной школе, которые связаны с внедрением ИКТ в практику учебной работы, с реальным уровнем использования и востребованности новых информационных технологий в образовательном процессе”. В рамках нового проекта предусмотрена регулярная (ежегодная) оценка изменения уровня компетентности школьников в области использования ИКТ в регионах проекта ИСО (Красноярский край, Республика Карелия, Пермская область, Челябинская область, Калужская область, Ставропольский край, Хабаровский край).

Давайте разберемся, что же собираются проверять в рамках этого проекта, и почему требуются какие-то новые инструменты для оценки определенных навыков и умений школьников, в данном случае — девятиклассников.

## 1. Детализация понятия ИКТ-компетентности

В рамках разрабатываемого проекта предполагается оценивать компетентность школьников в области владения ими информационно-коммуникационными технологиями (ИКТ). Обратите внимание, что в аббревиатуре “ИКТ” буква “К” соответствует слову *коммуникация*. В словаре иностранных слов указано: *коммуникация* — 1) путь сообщения (напр., воздушная к., водная к.); 2) форма связи (напр., телеграф, радио, телефон); 3) акт общения, связь между двумя или более индивидуумами, основанные на взаимопонимании; сообщение информации одним лицом другому или ряду лиц; 4) массовая к. — процесс сообщения информации с помощью технических средств... численно большим, рассредоточенным аудиториям. Таким образом, в рамках нового проекта будет проверяться способность учащихся (их компетентность в этой области) создавать и передавать информацию с использованием компьютерных технологий.

Что же стоит за термином “компетентность”, который начинает активно использоваться в отечественной педагогике и в официальных документах Министерства образования и науки? На сегодняшний день выработано

только лишь определение ИКТ-грамотности, и на базе этого определения вводится понятие ИКТ-компетентности.

**Определение.** *ИКТ-грамотность* — это использование цифровых технологий, инструментов коммуникации и/или сетей для получения доступа к информации, управления ею, ее интеграции, оценки и создания для функционирования в современном обществе.

Очевидно, что в данном определении используется несколько терминов и понятий, давайте последовательно рассмотрим каждое из них.

**ИКТ** — представление информации в электронном виде, ее обработка и хранение, но не обязательно ее передача. Информационно-коммуникационная технология представляет собой объединение информационных и коммуникационных технологий.

**Грамотность** — это динамичный инструмент (в самом широком смысле слова), позволяющий индивидууму постоянно учиться и расти.

**Цифровые технологии** относятся к компьютерному и программному обеспечению; **инструменты коммуникации** — к продуктам и услугам, с помощью которых передается информация, **сети** — это каналы передачи информации.

**Функционирование в современном обществе** отражает многообразие контекстов применения индивидуумом ИКТ-грамотности. ИКТ-грамотность предоставит индивидууму средства для успешной жизни и работы в экономически развитом или развивающемся обществе.

Введенное понятие ИКТ-грамотности определяет, какими же навыками и умениями должен обладать человек, чтобы его можно было назвать грамотным в данном смысле. Перечень этих навыков и умений приведен ниже в порядке повышения сложности познавательных (когнитивных) действий, необходимых для их выполнения:

- **определение** информации — способность использовать инструменты ИКТ для идентификации и соответствующего представления необходимой информации;
- **доступ** к информации — умение собирать и/или извлекать информацию;
- **управление** информацией — умение применять существующую схему организации или классификации;
- **интегрирование** информации — умение интерпретировать и представлять информацию. Сюда входит обобщение, сравнение и противопоставление данных;
- **оценивание** информации — умение выносить суждение о качестве, важности, полезности или эффективности информации;
- **создание** информации — умение генерировать информацию, адаптируя, применяя, проектируя, изобретая или разрабатывая ее;

• **передача** информации — способность должным образом передавать информацию в среде ИКТ. Сюда входит способность направлять электронную информацию определенной аудитории и передавать знания в соответствующем направлении.

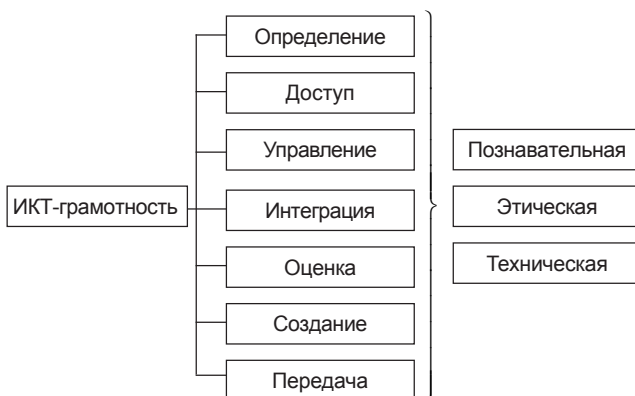
Перейдем теперь от определения ИКТ-грамотности к понятию ИКТ-компетентности. Компетентность — понятие очень общее. В нем не отражены конкретные познавательные-практические качества, которые должны быть сформированы в процессе обучения. Это связано с тем, что в разных областях профессиональной деятельности компетентность будет раскрываться с помощью различных познавательных-практических понятий. Поэтому в разных предметных областях вырабатываются свои определения компетентности, например, языковая компетентность.

Переход от “знаниецентрического” подхода в обучении (знания ради знаний) к “компетентностному” обучению предполагает воспитание такого человека и гражданина, который будет приспособлен к постоянно меняющимся условиям жизни. Пока круг компетенций, которым будут учить школьников, не определен окончательно, как и само понятие “компетентность”. Но за основу взяты способность брать на себя ответственность, участвовать в демократических процедурах, общаться и обучаться на протяжении всей жизни, проявлять самостоятельность в постановке задач и их решении. В рамках рассматриваемого проекта сформировано следующее определение ИКТ-компетентности.

**Определение.** Под *ИКТ-компетенцией* подразумевается уверенное владение учащимися всеми составляющими навыками ИКТ-грамотности для решения возникающих вопросов в учебной и иной деятельности, при этом акцент делается на сформированность обобщенных познавательных, этических и технических навыков.

Ниже приведена схематичная модель ИКТ-компетентности:

**Модель ИКТ-компетентности**



Для пояснения введенного понятия приведем примеры заданий, проверяющих навыки и когнитивные деятельности, составляющие ИКТ-компетентность.

- **Доступ** — найти и открыть соответствующее сообщение электронной почты в ящике входящих писем;
- **Управление** — найти и организовать соответствующую информацию из писем электронной почты;

- **Интеграция** — проанализировать преимущества и недостатки различных рекомендуемых чистящих средств для выведения пятен;

- **Оценка** — принять решение, какое средство для выведения пятен целесообразно заказать через интернет-магазин, основываясь на информации сайтов продавцов соответствующих товаров;

- **Создание** — представить свои рекомендации по решению некоторого вопроса в формате письма электронной почты.

## 2. Почему необходима разработка нового инструмента для оценки ИКТ-компетентности

Анализ существующих методов оценки компетентности школьников в области использования ИКТ, в том числе используемых в Российской Федерации в рамках эксперимента по единому государственному экзамену (ЕГЭ), а также рассмотрение подготовленных Министерством образования и науки РФ проектов стандартов по информатике (2004 года), показывают, что существующие методы оценки информационно-коммуникационной компетентности ориентируются почти исключительно на формирование технологических навыков и алгоритмического мышления школьников. Оценка умственных навыков высокого уровня, которые обеспечивают полноценную информационно-коммуникационную компетентность школьников, сегодня практически не обсуждается. Поэтому использовать существующие сегодня в РФ измерители компетентности школьников в области использования ИКТ (итоговая оценка по курсу информатики, или результаты ЕГЭ) не представляется возможным.

Одним из результатов процесса информатизации школы должно стать появление у учащихся способности использовать современные информационные и коммуникационные технологии для работы с информацией. Они должны уметь искать необходимые данные, организовывать, обрабатывать, анализировать и оценивать их, а также продуцировать и распространять информацию в соответствии со своими целями. Эта способность (или компетентность) должна обеспечить школьникам возможность:

- успешно продолжать образование в течение всей жизни (включая получение образовательных услуг с использованием Интернета);
- подготовиться к выбранной профессиональной деятельности;
- жить и трудиться в информационном обществе, в условиях экономики, которая основана на знаниях.

В связи с вышеизложенным было принято решение разработать новый инструмент оценки ИКТ-компетентности, который позволит, грубо говоря, проверить, как школьник мыслит и работает в “цифровом” мире. В рамках нового проекта оценка ИКТ-компетентности будет проводиться по результатам теста, выполняемого учащимся на компьютере, при этом общая ИКТ-компе-

тентность учащихся будет оцениваться по конечному результату. Составляющие навыки не будут выделяться и оцениваться индивидуально. Для оценки каждого из составляющих навыков в тесте будут представлены несколько заданий, но в целом за тест будет выставляться *качественная (диагностическая)* оценка, отражающая уровень ИКТ-компетенции: уровень выше базового, базовый, ниже базового.

Диагностическая оценка позволит исследовать познавательные и технические навыки, стоящие за конкретными составляющими ИКТ-компетенности. Результат такого типа тестирования будет полезен по нескольким причинам. Он может дать:

- общую оценку познавательной и технической компетенности ученика;
- более детализированную систему отчета о набранных баллах, выделяющую конкретные сильные и слабые стороны ученика;
- связь с существующими или специально разработанными методическими материалами;
- основываясь на результатах тестирования, можно выделять конкретные цели обучения.

Но, как известно, цель любой проверки, любой формы контроля достигается только на адекватно составленных (целям) вопросах и задачах. Одной из задач рабочей группы проекта является разработка заданий для теста. Без сомнения, тексты заданий публиковаться до тестирования не будут, но описать будущие задания можно и скорее всего необходимо.

1) Тест будет содержать 14 вопросов, выполнение каждого вопроса будет ограничено по времени. В тексте будет 1 вопрос на 30 минут, 2 вопроса на 15 минут и 11 вопросов на 5 минут.

2) Сценарий каждого задания будет приближен к реальной жизни. Например: *Используя программу поиска, найти сайты со статьями о болезнях сердца* (задание на оценивание умения “доступ”) или *В каждой из найденных статей найти информацию по вариантам лечения* (задание на оценивание умения “интеграция”). Если задание связано с каким-либо школьным предметом, то его содержание не выходит за рамки стандарта обучения по данному предмету. Попутно отметим, что в группу по разработке заданий входят учителя информатики, русского языка, химии и других предметов.

3) Для выполнения заданий теста разрабатывается специальная тестирующая среда, включающая имитацию программы поиска в Интернете, почтовой программы, текстового и графического редакторов и т.д. Это делается для того, чтобы уравнивать условия выполнения теста учащимися (ранее полученные навыки работы с каким-либо программным средством не дадут большого преимущества перед теми ребятами, которые по объективным причинам таких навыков в школе приобрести не смогли).

Ниже приведен пример вопроса на 30 минут (автор — учитель словесности средней общеобразовательной школы № 542 Т.Н. Козлова).

### 3. Надо ли специально готовить школьников к будущему тестированию по ИКТ-компетенности?

Одной из косвенных целей данного проекта является акцентирование внимания учителей всех профилей (не только информатики) на развитие у учащихся навыков, составляющих ИКТ-компетенность. Именно поэтому важным было выработать определение ИКТ-грамотности и ИКТ-компетенности. В дальнейшем необходима разработка методик выработки навыков, составляющих ИКТ-компетенность, и учебных заданий, на основе которых эти навыки можно будет выработать. На сегодняшний момент для каждого учителя важно понять, что входит в ИКТ-компетенность, проанализировать, какие учебные задачи можно модифицировать так, чтобы косвенно они работали на выработку навыков, составляющих ИКТ-компетенность.

По мере работы рабочей группы проекта “Оценка ИКТ-компетенности девятиклассников” на страницах газеты “Информатика” будут публиковаться новые материалы, которые, как мы надеемся, будут интересны и полезны учителям информатики. Кроме того, в 2006 году в рамках дистанционных курсов повышения квалификации для учителей информатики будет прочитан курс с условным названием “Как готовиться к тестированию по проверке ИКТ-компетенции школьников”.

### 4. Пример задания. “Выдающиеся русские лингвисты”

*Пояснения для учителей.* В курсе русского языка 9-го класса перед итоговым повторением учащимся предлагается тема “Общие сведения о языке”. В рамках этой темы учащимся рассказывается о выдающихся русских лингвистах (в соответствии с образовательным стандартом по русскому языку за курс основной школы). Эти сведения позволят учащимся:

- сформировать научное представление о русском языке;
- понять, что орфография и пунктуация являются лишь частью предмета;
- оценить вклад выдающихся русских лингвистов не только в русскую, но и мировую лингвистику.

Данное задание с точки зрения ИКТ-компетенности нацелено на выявление навыков по поиску и оценке информации. Компетенность выявляется с помощью задания оценить информацию из базы данных с использованием предоставленного механизма поиска, выбрать те ресурсы, которые можно использовать для справки, или те, которые отвечают определенным требованиям. Время на выполнение: 30 минут. Выявляемые навыки:

- Доступ;
- Управление;
- Интеграция;
- Оценка;
- Создание текстового документа.

Данное задание является характерным примером проверки сочетания технологических и когнитивных навыков.

ИКТ-компетентности	Сценарий для учащегося	Примерные действия учащегося
Доступ	Используя программу поиска, найти сайты со статьями о <b>русских лингвистах</b> (языковедах, русистах)	В поисковой системе учащийся вводит термин поиска "русские лингвисты" или сразу вводит фамилию ученого
Управление	Оценить сайты и выделить те, из которых можно получить необходимую информацию о каком-либо <b>одном</b> ученом	Выбор сайтов 1. <a href="http://school.edu.ru">http://school.edu.ru</a> 2. <a href="http://pilic.narod.ru">http://pilic.narod.ru</a> (Сайт № 2 наиболее информативен, т.к. содержит общую информацию по теме теста с указанием <b>имен ученых</b> . Среди них кандидат должен выбрать одного лингвиста.)
Интеграции	В отобранных статьях найти информацию о <b>биографии ученого</b> , его <b>научной деятельности</b> (ее основное направление)	В поисковой системе учащийся вводит термин поиска (например, "Л.В. Щерба" "биография Л.В. Щербы") Выбор сайтов 1. <a href="http://www.ruthenia.ru/apr/textes/sherba/bio.htm">http://www.ruthenia.ru/apr/textes/sherba/bio.htm</a> 2. <a href="http://www.philot.msu.ru/rus/galya-1/bibl/jsh.htm">http://www.philot.msu.ru/rus/galya-1/bibl/jsh.htm</a>
Оценка	Сравнить информацию из разных статей, выделить <b>разные аспекты деятельности лингвиста и сведения о его жизни</b>	Чтение текста. Отбор информации
Создание	Разработать документ в текстовом редакторе — конспект статьи на лингвистическую тему о <b>выдающемся русском лингвисте</b>	Создание текстового документа. Примерный план итогового документа: 1. Даты жизни, место рождения, учебы, интересы. 2. Где работал, принадлежность к лингвистической школе (Московской, Петербургской, Казанской). 3. Вклад в лингвистическую науку. 4. Основные научные труды

## Начисление баллов

Наблюдаемые данные	Результат работы	Уровень	Методы	Соответствие стандартам образования
Качество синтаксиса	Термины поиска	Высокий Средний Низкий	Использует "И" при первом поиске в сети. Сразу вводит имена лингвистов. Не использует "И" при первом поиске, но использует при последующем. Не использует "И"	Знание видных ученых-русистов, исследовавших русский язык
Качество и уровень работы с выбранными ресурсами	Выбранные ресурсы	Высокий Средний Низкий	Все выбранные ресурсы получают 5 баллов за весомость, объективность, охват материала. Объем чтения — 150–170 слов в минуту Свыше 80, но менее 100% выбранных ресурсов получают 5 баллов за весомость, объективность, охват материала. Объем чтения — 120–140 слов в минуту Менее 80% выбранных ресурсов получают 5 баллов за весомость, объективность, охват материала. Объем чтения — менее 120 слов в минуту	Умение быстро и осознанно читать текст
Качество полученного текста	Осознанность в отборе материала; логичность; стиливое соответствие языка	Высокий Средний Низкий	Соблюдена логическая последовательность в изложении материала; вычленены все микротемы исходного текста; сохранено стиливое единство текста. В целом соблюдена логическая последовательность; вычленены свыше 80, но менее 100% микротем исходного текста; сохранено стиливое единство текста; наличие 1–2 речевых ошибок. Не соблюдена логическая последовательность в изложении материала; вычленены менее 80% микротем исходного текста; отбор языковых средств нарушает стиливое единство текста; наличие более двух речевых ошибок	Умение составлять конспект статьи на лингвистическую тему; совершенствовать содержание и языковое оформление в своем тексте



## “Как это делаю я”

# Методический конкурс для учителей информатики

### Задание пятого тура

Газета “Информатика” проводит в 2005/2006 учебном году методический конкурс для учителей информатики “Как это делаю я”.

**Цель конкурса:** обмен опытом решения повседневных методических задач.

**Суть конкурса:** для каждого тура конкурса (всего в 2005/2006 учебном году будет проведено 6 туров) предлагается типичная методическая задача, которую часто приходится решать учителю информатики. В зависимости от предлагаемой задачи указывается формат представляемых на конкурс материалов. По завершении каждого тура лучшие работы появляются на страницах газеты. На страницах газеты также публикуется список всех участников данного тура. Все присланные на конкурс материалы размещаются на сайте “Информатики” <http://inf.1september.ru>. Авторы лучших работ получают призы, предоставленные редакцией и нашими друзьями.

**Ключевая особенность** нашего конкурса заключается в том, что и задания, и форма предоставления конкурсных материалов фиксированы. Это позволяет с достаточной высокой степенью объективности сравнить материалы участников и отметить лучшие.

Кроме того, необходимо отметить, что соревновательная составляющая нашего конкурса, конечно, является вторичной. Мы уверены, что всем нам будет просто интересно и полезно обменяться опытом решения типичных проблем, которые возникают перед каждым из нас.

### ЗАДАНИЕ ПЯТОГО ТУРА

#### “Кабинет — наш второй дом”

Оформление кабинета информатики — не только эстетический (что, конечно, также немаловажно), но и методический вопрос. Как вы его решаете? Имеются ли у вас стенды? Как они оформлены? Каким образом наглядные материалы на стендах применяются в текущей учебной деятельности?

Мы знаем, что одними только стендами дело часто не ограничивается. Каких только замечательных кабинетов информатики мы не видели! И с росписями (содержательными!) на стенах, и с “встроенными” музеями вычислительной техники, и с другими неожиданными элементами оформления.

Пожалуйста, поделитесь с коллегами своим опытом. Нас интересуют находки любого масштаба (музеи и росписи — это, конечно, здорово и “роскошно”, но

иногда и один небольшой стенд, сделанный вдумчиво и аккуратно, не менее полезен).

**На конкурс** требуется представить описание наглядных элементов оформления вашего кабинета. Описание в обязательном порядке должно быть снабжено иллюстративным материалом (фотографиями).

#### Формат представления конкурсных материалов

Материалы принимаются только в электронном виде. Текстовая часть (описание) должна быть представлена в файлах формата Microsoft Word (любой версии). Объем текста (в знаках, с учетом пробелов) — не более 5000 знаков. Иллюстративный материал должен быть представлен в файлах формата JPEG, с разрешением не менее 800 × 600 пикселей, сохраненных с качеством не хуже среднего.

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](mailto:inf@1september.ru).

В теме (subject) электронных писем, пожалуйста, указывайте “Методический конкурс, пятый 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 пятого тура — до 28 февраля 2006 г. (для “бумажных” писем дата отправки фиксируется по штемпелю предприятия-отправителя).

Результаты пятого тура будут опубликованы в № 8/2006. На сайте “Информатики” материалы будут размещаться по мере поступления.

Задание шестого тура будет опубликовано в № 4/2006.

## “Как это делаю я”

# Методический конкурс для учителей информатики

### Итоги второго тура

Дорогие коллеги! Настало время подвести итоги второго тура нашего конкурса. Его темой было соблюдение техники безопасности. Тему трудно причислить к интересным и творческим, но ее не обойти: все учителя информатики должны в обязательном порядке знакомить учеников с правилами техники безопасности при работе в компьютерном классе. Однако как добиться от учеников их реального соблюдения? Как донести до ребят то, что сухие строчки инструкций не являются пустой формальностью, что соблюдать их в буквальном смысле слова жизненно необходимо? Своим опытом решения указанного вопроса с нами поделилось 34 участника. Все присланные материалы размещены в разделе конкурса на сайте “Информатики” <http://inf.1september.ru>.

Для публикации в газете мы выбрали несколько материалов:

БЕРКУТОВОЙ Галины Алексеевны, СОШ № 64 (г. Красноярск);

ЗАГВОЗДИНОЙ Любви Зионовны, СОШ № 14 (г. Братск, Иркутская обл.);

ЛЕОНОВОЙ Елены Семеновны, гимназия № 11 (г. Тула);

ТРОФИМОВОЙ Веры Владимировны, МНОУ “ЛИЦЕЙ” (г. Кемерово);

ЧЕРНОБАБОВОЙ Клавдии Владимировны, гимназия № 8 (г. Сочи);

ЯМКИНОЙ Елены Владимировны, СОШ № 37 (г. Ульяновск).

В основном это фрагменты работ, поскольку нет необходимости представлять на страницах газеты журналы по технике безопасности или формулировку самих правил.

К сожалению, Интернет доступен не для всех участников, так что часто возникает вопрос: “Как в таком случае можно познакомиться с работами, присланными на конкурс?”. Выход был найден: в конце учебного года все, кто примет участие хотя бы в одном туре конкурса, получат специальный компакт-диск с работами участников за все туры. Напомним, что всего предполагается провести шесть туров; задание четвертого тура (крайний срок отправки материала — 31 января 2006 г.) было опубликовано в № 24/2005 и на сайте “Информатики”.

#### Участники второго тура

Александренкова М.В., гимназия № 1527 (г. Москва);

Аристова Н.А., средняя русско-татарская школа № 17 (г. Альметьевск, Республика Татарстан);

Беркутова Г.А., СОШ № 64 (г. Красноярск);

Беспалько Н.Т., гимназия № 3 (г. Инта, Республика Коми);

Брагина А.А., СОШ (с. Антушево, Вологодская обл.);  
Брылев С.В., СОШ № 3 им. героя РФ А.Н. Епанешникова (г. Елабуга);

Володина Н.Ф., Межшкольный учебный комбинат № 3 (г. Комсомольск-на-Амуре);

Волошинская Е.Л., СОШ № 145 (г. Красноярск);  
Воронкова О.Б., Суворовское училище МВД России (г. Новочеркасск);

Ганиева Л.Р., СОШ (с. Трунтаиш, Республика Башкортостан);

Гоняева Т.А., СОШ (с. Зерновое, Иркутская обл.);  
Жуланова С.В., СОШ № 3 (г. Березники, Пермская обл.);

Загвоздина Л.З., СОШ № 14 (г. Братск, Иркутская обл.);

Исаева Л.В., СОШ (дер. Большие Калмыки, Тульская обл.);

Исакова И.С., СОШ № 1 (п. Лимбьяха, г. Новый Уренгой);

Искандарова А.Р., СОШ № 18 (г. Уфа, Республика Башкортостан);

Кодесникова Н.В., Чулымская СОШ (п. Чистое Поле, Красноярский край);

Королев Ю.Н., лицей им. Н.И. Лобачевского при КГУ (г. Казань, Республика Татарстан);

Костарева Н.П., СОШ № 3 (г. Оса, Пермская обл.);  
Леонова Е.С., гимназия № 11 (г. Тула);

Назарова И.В., СОШ № 13 (г. Оленегорск, Мурманская обл.);

Обухова И.В., СОШ № 3 (п. Октябрьский, Иркутская обл.);

Орлова Е.В., СОШ № 1 (г. Стерлитамак, Республика Башкортостан);

Перминова О.И., СОШ № 21 (г. Канск, Красноярский край);

Севрюкова Н.Н., СОШ № 2 (с. Богучаны, Красноярский край);

Семенова З.С., СОШ № 7 (г. Заинск, Республика Татарстан);

Трофимова В.В., МНОУ “ЛИЦЕЙ” (г. Кемерово);  
Урдиханова И.Н., СОШ № 2 им. академика А.И. Берга (г. Жуков);

Цыганкова Е.В., СОШ (с. Михайловка, Курская обл.);  
Чапкевич И.М., лицей № 4 (г. Орел);

Чернобабова К.В., гимназия № 8 (г. Сочи);  
Шеломенцева Л.В., СОШ (с. Олекан, Читинская обл.);

Широкова Л.В., лицей № 10 (г. Волгоград);  
Ямкина Е.В., СОШ № 37 (г. Ульяновск);

Яшина Е.Ю., гимназия № 5 (г. Давлеканово, Республика Башкортостан);

\* Конкурсные материалы подвергаются минимальному редактированию.

## “Соблюдаем технику безопасности” Материалы участников конкурса

### Общий план урока “Техника безопасности”

Е.С. Леонова,  
г. Тула

Уважаемые коллеги! Предлагаю вашему вниманию общий план урока “Техника безопасности”.

Принято считать, что это — одна из самых скучных тем. Почему?

Я считаю, причина в том, что подавляющее большинство педагогов по-прежнему ограничивают ее кругом вопросов “Что можно и чего нельзя делать в компьютерном классе”. У современных “продвинутых” детишек это, как правило, вызывает реакцию типа: “Ну, вот, опять...”

И их можно понять: как обращаться с утюгом — сегодня большинство знает. В этом смысле компьютер постепенно становится такой же привычной бытовой техникой (как самостоятельно, так и вместе с разнообразной цифровой периферией). Раньше “Правила техники безопасности” дети слушали, затаив дыхание, так как сам кабинет вызывал у них священный трепет. Сегодня для большинства это привычная, почти домашняя среда, и, конечно, здесь они твердо знают “как надо и как не надо”.

Учитывая это, я в последние лет пять посвящаю вопросам безопасности целых два урока, при этом сами правила техники безопасности не “вдалбливаются” мною напрямую, а ненавязчиво и естественно продолжают и дополняют тему охраны здоровья и хорошего самочувствия моих подопечных, а что может быть важнее этого?

#### 1. Беседа в виде вопросов-ответов

Чем отличается изображение на бумаге и на экране? (Экран является источником света с активной контрастностью, в отличие от листа бумаги с пассивной контрастностью.)

Каким должно быть расстояние от глаз до монитора, чтобы меньше уставать?

Какой монитор — ЖК или ЭЛТ — меньше утомляет зрение?

Влияние какого излучения не чувствует глаз, но чувствует нервная система в целом, так как оно воздействует на кожу и общее самочувствие?

Показ ролика на CD “Эргономика рабочего места”.

Предлагаю детям темы для доклада: “Как работает ЖК-монитор” и “Устройство ЭЛТ-монитора”.



Рассказываю о синдроме запястного канала, о судебных процессах (в 1995 г. в одном только Нью-Йорке их было три) по поводу синдрома запястного канала истцов, и каждый раз выигрывали ответчики: компании IBM, Apple и др., менеджеры которых утверждали, что самое эффективное средство профилактики — “теоретическое и практическое обучение персонала”, т.е. ваше, ребята, обучение. Показываю, какими должны быть правильные (эргономичные) клавиатуры, определяем с ребятами оптимальное расстояние от края стола до клавиатуры. Говорим о видах излучения от компьютера в целом и от его отдельных блоков.

#### 2. Компьютеры и экология

Подвожу ребят к мысли о том, что нет экологии вообще, как нет здоровья вообще. Экология на планете начинается с умения поддерживать порядок на своем рабочем месте (включая привычку ходить в чистой обуви).

**Задание на дом** (из цикла “Вредные задачи”, посвящается Г.Остеру).

№ 1. Секретарша Ирочка очень боится излучения от монитора. Защитный экран понижает его в два раза. Сколько защитных экранов придется уста-

новить, если для себя Ирочка решила снизить изучение в 32 раза?

№ 2. На дискете было 2 плохих блока. После того как Кирюша бросил дискету в суп и дал ее облизать хомяку Кольке, плохих блоков стало в 5 раз больше. Сколько будет искать эту дискету с важными файлами папа, пока не догадается, что хомяк Колька не ограничился облизыванием дискеты?

## Изучаем и повторяем правила техники безопасности

**В.В. Трофимова,**  
п. Октябрьский, Иркутская обл.

Правила ТБ имеются в каждом кабинете информатики. Учитель должен проводить с детьми соответствующий инструктаж перед началом учебного года. Только как же скучно это звучит: правила, инструктаж... Нам и самим обычно хочется быстрее закончить эти необходимые формальности, а уж дети-то зевают, не стесняясь. При этом все понимают, что правила эти, конечно, важные, соблюдать их следует вовсе не для галочки, а именно ради собственной безопасности.

При работе с маленькими детьми (учениками 1–5-х классов) ситуация усугубляется. Старшие дети способны какое-то время уделять внимание даже скучной и неинтересной теме, с малышами так не получится. Если уж неинтересно, то удержать их внимание можно лишь административными методами, что не слишком эффективно. Зато, если детям интересно, то даже сухие и скучные правила техники безопасности запоминаются и, главное, *понимаются*.

При изучении правил ТБ я использую стихи, любезно сочиненные по моей просьбе коллегой — учителем русского языка О.В. Куркиной. В качестве основы взяты правила, изложенные в книге А.А. Дуванова “Азы информатики” (книга 1 “Знакомство с компьютером”), пиктограммы в виде запрещающих знаков расположены на стенде, а стихи дети с удовольствием запоминают и в дальнейшем легко воспроизводят.

Вы вошли впервые в класс,  
Где компьютеры стоят.  
Необычен класс для вас,  
Он особый, говорят.

Ведь компьютеры — машины,  
И пришли вы не играть.  
Правила необходимо  
Здесь вам строго соблюдать.

1. Чтоб работали исправно  
Все компьютеры всегда,  
Нужно помнить вам о главном:  
Главное здесь — чистота.

Пыль компьютеру — лишь враг,  
И, отправившись сюда,  
Вычисти портфель, пиджак.  
Для машины грязь — беда.

Будут крупные проблемы:  
Портит пыль электросхемы!

2. Вот, обедая в столовой,  
Вы услышали звонок...  
Это вам уже не ново —  
Мчитесь с булкой на урок.

Стоп! Сюда вам хода нет!  
Хоть и булочки вкусны,  
Здесь недопустим обед,  
Здесь — рабочие столы!

Крошки в клавишах порой  
И влекут в работе сбой.

3. Если липки ваши руки  
От несъеденных конфет,  
Вы не трите их о брюки,  
Направляясь в кабинет.

И жевательной резинке,  
Просим помнить вас, друзья,  
Место в мусорной корзинке,  
Здесь жевать ее нельзя.

Что испачкано, порой  
Отмывать нельзя водой!

4. Нужно помнить всем вам, дети,  
То, что тоже важно знать:  
Дисциплину в кабинете  
Нужно строго соблюдать.

А крикливым шалунишкам,  
Тем, что любят поиграть,  
Здесь не место. Много слишком  
Можно невзначай сломать!

Всюду здесь аппаратура:  
Монитор, клавиатура.

5. И хотелось бы заранее  
Дать наказ еще такой,  
Что нельзя без указаний  
Кнопки трогать ни одной!

Не касайтесь монитора —  
Пятна сразу здесь видны.  
Не хватайтесь для опоры  
Вы за электрошнуры.

С напряжением в сети  
Ох, дружок, не шути!

Не сложны советы эти  
 Вроде бы для всех вас, дети,  
 Но у самого порога  
 Вспомни их, им следуй строго.  
 И компьютер — добрый друг  
 Поведет вас в мир наук!

При повторении правил ТБ часто использую проблемные ситуации. Например, такие:

1. Знайка набирал на компьютере длинный и сложный текст, а Незнайка сидел рядом и наблюдал за его действиями. Внезапно он изо всех сил стал трясти Знайку за плечо и со всего размаха ткнул пальцем в монитор так, что последний начал вибрировать.

— Ты совершил ошибку! — закричал Незнайка. — Это слово пишется по-другому!

Знайка посмотрел на Незнайку сквозь толстые линзы своих очков и...

*Как вы думаете, что в ответ сказал Знайка?*

2. Незнайка пришел в кабинет вычислительной техники. Его внимание привлекло огромное количество питающих проводов, которые располагались с тыльной стороны каждого монитора. Решив рассмотреть их поближе, он уже протянул руку к одному. Внезапно кто-то схватил его за плечи и изо всех сил потянул назад. Кем-то оказался Знайка.

*Почему Знайка повел себя таким необычным образом?*

3. Винни-Пух копал картошку, а Пятачок ее собирал. Опустив в погреб последний мешок, они сразу же отправились на занятия в компьютерный класс.

*Что они забыли сделать?*

Незнайка работал на компьютере. Внезапно экран погас. Не зная, что делать, он стал изо всех сил нажимать на клавиши, то выключать, то включать компьютер.

*Правильно ли он поступил?*

С удовольствием работают ребята и с сигнальными карточками (не только при изучении и повторении правил ТБ). Правила такие. Перед учениками находятся карточки красного и зеленого цветов. В ходе игры учитель читает правила ТБ. Если очередное предложение действительно является правилом ТБ, то ребята поднимают зеленую карточку. По аналогии с правилами дорожного движения это означает, что ребята разрешают учителю “двигаться дальше” — читать следующее “правило”. Если же то, что произнес учитель, не будет являться правилом ТБ, то дети поднимают красную карточку. Она является сигналом, что учитель должен остановиться и обсудить, что именно сказано неверно.

## Знакомство с правилами техники безопасности

Л.З. Загвоздина,  
 г. Братск

Уроки информатики в компьютерном классе я начинаю со знакомства с правилами техники безопасности в кабинете. Говорю учащимся о том, что компьютерный класс, в котором к каждому рабочему месту подведено опасное для жизни напряжение, является местом повышенной опасности. Затем знаколю с правилами.

На каждом уроке напоминаю правила поведения в кабинете информатики. После знакомства с правилами предлагаю учащимся нарисовать дома плакат по ТБ, затем вместе с учениками подводим итоги конкурса плакатов; лучшие вывешиваются на стенд по технике безопасности\*.

На уроке обязательно проводятся физкультминутки для профилактики утомляемости. В младших классах в течение каждой четверти проводится игра “Золотой ключик”.

Правила игры следующие: на плакате расположена лестница со ступеньками с двух сторон. С одной стороны по ступенькам (их столько, сколько уроков в четверти) поднимается Буратино (это ученики), а по другой — Карабас Барабас. Игрокам нужно первыми добраться до Золотого ключика, а для этого необходимо не нарушить на уроке правила ТБ. Если правила нарушены, то на ступеньку выше поднимается Карабас Барабас. Ребята могут победить Карабаса Барабаса, только если быстрее доберутся до Золотого ключика.

## Инструктаж по технике безопасности

Е.В. Ямкина,  
 г. Ульяновск

Инструктаж по технике безопасности проводится раз в полугодие первыми уроками. На самом первом уроке диалог направляется в такое русло: “Здравствуйте, ребята. Вы пришли на урок в компьютерный класс. В кабинете установлена дорогостоящая техника, к рабочим местам учащихся и учителя подведено электричество. Как вы думаете, с какой темы урока должно начаться наше с вами общение? Какая тема у нас с вами будет повторяться из года в год, и в 8-м, и в 9-м, и даже в 10–11-м классах? И не один раз? Правильно, правила по технике безо-

\* См. в рубрике “На стенд в кабинете информатики” на с. 25.

пасности! А как вы думаете, только в школе на уроках проводится инструктаж по ТБ? А на производстве? А у кого родители проходили хоть раз инструктаж? Спросите у них, как это происходит? Какая, по-вашему, ответственность ложится на того, кто инструктирует? А на того, кого инструктируют? А вообще, в чем смысл инструктажа?”

Получив ответы на поставленные вопросы, обсуждаем с учениками, какая техника находится в кабинете и какое назначение она имеет. В конце обязательно показываю ученикам щиток, объясняю, что к нему подходит только учитель, но в экстренных ситуациях они должны знать кнопку общего

рубильника: учитель может оказаться или далеко от нее, или не иметь к ней доступа.

Далее учащиеся читают саму инструкцию и расписываются в журнале.

Сев за компьютеры, проходят шуточный тест по ТБ, за который получают оценку.

На последующих инструктажах учащимся предлагается посмотреть презентацию по ТБ: что можно, а что нельзя делать в компьютерном классе, пройти тестирование.

В старших классах ученики сами готовят соответствующие презентации за отведенное на инструктаж время.



## Соблюдаем технику безопасности

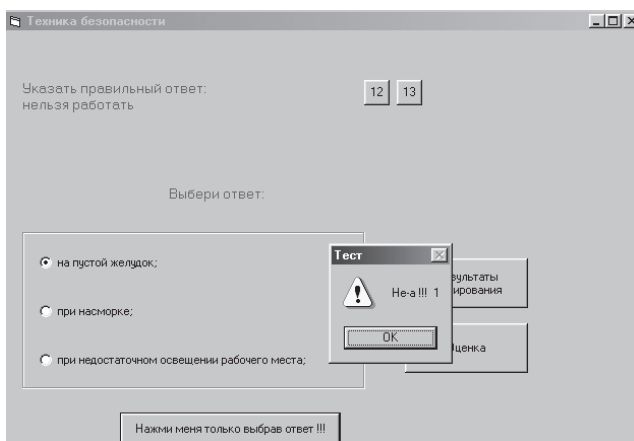
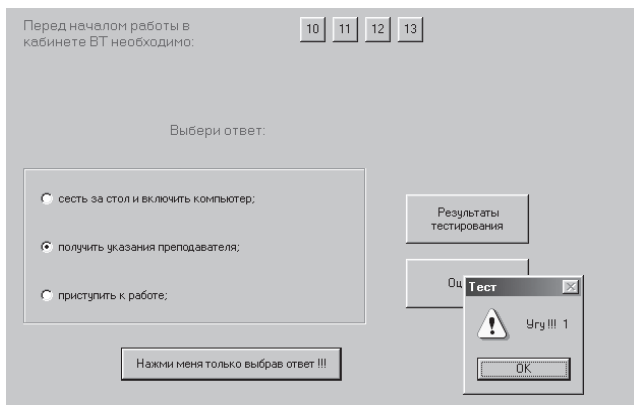
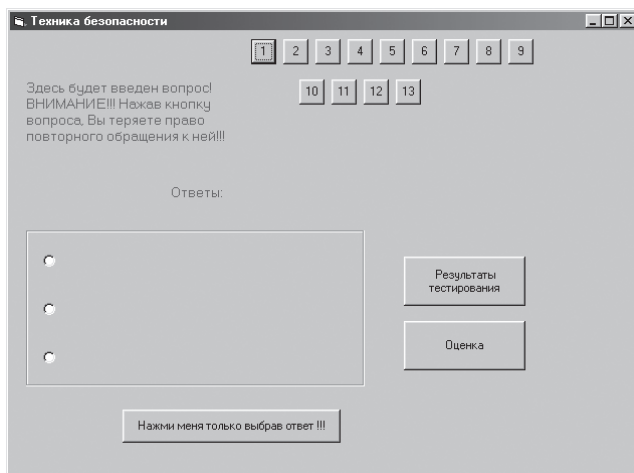
Г.А. Беркутова,  
г. Красноярск

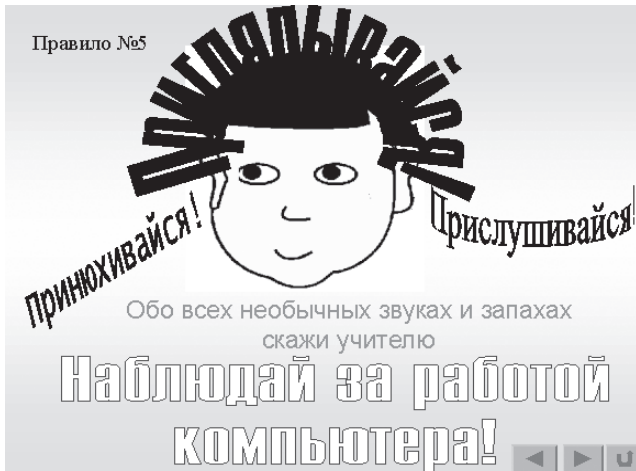
Не секрет, что ученики младших классов намного лучше воспринимают и запоминают информацию в сопровождении зрительных образов: картинок, макетов и т.п. Плакаты по ТБ в какой-то степени помогали, но они не затрагивали эмоциональных струн детской души, да и внешнее оформление плакатов проигрывало перед картинками, предлагаемыми детям компьютерными технологиями с их яркими, насыщенными цветами и анимацией. Тогда и возникла идея создать свои яркие, запоминающиеся наглядные пособия.

В качестве таких наглядных пособий я решила сделать учебный игровой фильм и интерактивную, с анимационными эффектами презентацию по ТБ.

Фильм получил название “Мишкин сон”. Главный герой фильма абсолютно не соблюдает правила ТБ, и в фильме обыгрываются ситуации, которые из-за этого с ним происходят. Чтобы фильм не получился “страшилкой”, для снятия напряжения у детей все ситуации в фильме подаются с определенной долей юмора. Думаю, что сделали мы это не зря, так как дети младшего возраста воспринимают события фильма почти как реальные.

Презентация была создана для того, чтобы ученики могли подробнее узнать о факторах, влияющих на организм человека, закрепить усвоение правил, законспектировать их и при этом поработать на компьютере. Младшие дети правила не конспектируют, но с удовольствием “бродят” по презентации. В презентации имеется 12 слайдов с описанием правил и 3 слайда (для любознательных), в которых рассказывается, почему вреден тот или иной фактор. Переход на эти слайды осуществляется по гиперссылкам и с помощью управляющих кнопок  и , что дает возможность ученику не просто пассивно смотреть на экран, но и принять участие в





управлении презентацией. Кроме того, в начало презентации помещен структурный слайд, предоставляющий произвольный выбор любого из правил. Переход между слайдами установлен не по времени, а по щелчку мыши, чтобы каждый ученик мог работать в собственном темпе.

## Последствия несоблюдения правил ТБ

### При несоблюдении правил вы рискуете здоровьем

Если у вас пока еще хорошее зрение, то в через несколько лет вы будете в очках, если уже плохое, то придется покупать бинокль; спина согнется, приобретете множество болезней, а выглядеть будете вот так

Ну и как нравится? Нет?

Тогда возьмите ручку, откройте тетрадь, и запишите все правила.

Тем более, что на следующем уроке придется отвечать их на оценку.

**Ни пуха, ни пера!**



Результат проведения уроков с такими наглядными пособиями превзошел все ожидания: после просмотра фильма равнодушных не осталось, дети обсуждали события фильма, отвечали на вопросы по правилам и активно работали с презентацией.

## Соблюдаем технику безопасности

К.В. Чернобава,  
г. Сочи

Урок по технике безопасности в компьютерном кабинете проводить необходимо. Этого требуют и нормативные документы образовательных программ, и здравый смысл: компьютерный класс, в котором к каждому рабочему месту подведено опасное для жизни напряжение, действительно является местом повышенной опасности.

В нашем кабинете имеются: “Инструкция по охране труда учащихся в компьютерном кабинете”, утвержденная директором (на рабочем стенде); шуточный стенд “Не повторяй это!”.

Мой личный опыт показал, что на строгую “Инструкцию по охране труда учащихся в компьютерном кабинете”, подписанную директором, ребята обычно внимания не обращают. Существовала когда-то практика иметь инструкцию по ТБ на каждом рабочем месте ученика, но она тоже особого интереса не вызывала.

А вот шуточный плакат привлекает внимание ребят: они подходят, читают, делятся впечатлениями, смеются, что-то обсуждают.

Чтобы убедить ребят в том, что сухие строчки правил не являются пустой формальностью, что соблюдать правила техники безопасности жизненно необходимо, и был придуман этот стенд.

Хотелось бы поблагодарить разработчиков “Руководства пользователя ПК” фирмы “Формоза-Центр” и, в частности, художника Д.А. Бурусова: именно он



создал шуточный образ человека, действия которого повторять строго запрещается.

В дальнейшем шуточный герой с плаката переключал в презентацию, созданную вместе с ребятами\*.

Теперь урок стал электронным. Он включает: лекционный материал, сопровождаемый яркими слайдами с мультимедийными эффектами. Кино на большом экране усиливает эмоциональное, психологическое, интеллектуальное восприятие материала, видеоролик с CD-диска “Мир информатики”, самостоятельную работу с электронным материалом. После рассказа-лекции и обсуждения (здесь ребята делятся своими наблюдениями и случаями из жизни) каждый может еще раз просмотреть презентацию на своем рабочем месте и сделать конспект того, что лично ему может пригодиться, чего он еще не знает.

На следующем уроке проводится самостоятельная работа в форме теста.

\* См. в рубрике “На стенд в кабинете информатики” на с. 24.



Запрещается подсоединение и отсоединение интерфейсных кабелей при работающем компьютере



Разместить составляющие компьютера нужно близко друг от друга, чтобы хватало длины соединительных кабелей



Не касайтесь электрических контактов, а также не просовывайте никаких предметов через вентиляционные отверстия корпуса

**Соблюдаем технику безопасности**



Избегайте мест с прямым попаданием солнечных лучей, избыточно влажных и пыльных



После установки компьютера на рабочем месте его следует надежно заземлить.



Обеспечение циркуляции воздуха в зоне вентиляционных отверстий, предотвращающих перегрев системного блока.

**Знакомство с правилами техники безопасности**

**ПРАВИЛА ПОВЕДЕНИЯ В КАБИНЕТЕ ИНФОРМАТИКИ**







# Системы счисления и их применения\*

С.Б. ГАШКОВ,

Москва

## § 1. Деньги в конвертах и зерна на шахматной доске

Представьте себе, дорогой читатель, что вы банкир, занимающийся отмыванием грязных денег, и завтра ждете важного клиента, которому вы должны выдать круглую или не очень круглую, но заранее вам неизвестную сумму от 1 до 1 000 000 000 у. е. Чтобы не пачкать руки о грязные деньги, вы заранее дали указание своим кассирам заготовить некоторое количество конвертов с деньгами, на которых написаны содержащиеся в них суммы, и собираетесь просто отдать клиенту один или несколько конвертов, в которых и будет содержаться требуемая им сумма. Какое наименьшее количество конвертов необходимо иметь?

Конечно, можно просто заготовить конверты со всеми суммами от 1 до 1 000 000 000. Но где взять столько денег на конверты?

1. А какова будет в этом случае полная сумма во всех конвертах? Попробуйте оценить также массу бумаги, предполагая, что использованы не более чем сотенные купюры<sup>1</sup>.

Есть более рациональный подход к нашему делу. Надо положить в первый конверт 1 у. е., а в каждый следующий класть вдвое большую сумму, чем в предыдущий. Тогда, например, в 5-м конверте будет 16 у. е., в 10-м — 512 у. е., в 11-м — 1024 у. е., в 21-м —  $1024^2 = 1\,048\,576$  у. е., в 31-м —  $1024^3 = 1\,073\,741\,824$  у. е., но он нам, очевидно, уже не понадобится, а вот 30-й с  $1\,073\,741\,824/2 = 536\,870\,912$  у. е. может и пригодиться. В общем случае сумма в  $(n+1)$ -м конверте будет равна произведению  $n$  двоек, это число принято обозначать  $2^n$  и называть  $n$ -й степенью двойки. Условимся считать, что  $2^0 = 1$ . Проведенные выше вычисления основывались на следующих свойствах операции возведения в степень:

$$2^n 2^m = 2^{n+m}, \quad 2^n / 2^m = 2^{n-m}, \quad (2^n)^m = 2^{nm}.$$

Экспериментально легко проверить, что любое число можно представить единственным образом в виде суммы различных меньших степеней двойки, и поэтому наша задача почти решена. Например,

$$30\,000 = 2^{14} + 2^{13} + 2^{12} + 2^{10} + 2^8 + 2^5 + 2^4.$$

Но для реального применения нужен алгоритм построения такого разложения. Далее будут приведены несколько разных алгоритмов, но вначале мы рассмотрим самый простой. В сущности, это алгоритм выдачи сдачи клиенту, записанный некогда даже в инструкции для работников торговли, но очень редко ими выполняющийся. А он очень прост — сдачу надо выдавать, начиная с самых больших купюр. В нашем случае нужно найти конверт с наибольшей суммой денег, не превосходящей требуемую, т.е. наи-

большую степень двойки, не превосходящую требуемого количества денег. Если требуемая сумма равна этой степени, то алгоритм заканчивает работу. В противном случае опять выбирается конверт с наибольшей суммой денег, не превосходящей оставшуюся, и т.д. Алгоритм закончит работу, когда останется сумма, в точности равная степени двойки, и она будет выдана последним конвертом.

Ниже мы докажем, что, имея набор конвертов с суммами в 1 у. е., 2 у. е., 4 у. е., ...,  $2^n$  у. е., любую сумму денег от 1 у. е. до  $2^{n+1} - 1$  у. е. можно выдать единственным способом. Также будет доказано, что, действуя по описанному алгоритму, мы всегда получим этот способ выдачи требуемой суммы.

Вначале рассмотрим пример работы алгоритма с числом  $2^n - 1$ . Ясно, что на первом шаге будет выбрано число  $2^{n-1}$ , останется число  $2^n - 1 - 2^{n-1} = 2^{n-1} - 1$ , потом будет выбрано число  $2^{n-2}$ , и т.д., и в результате получится разложение

$$2^n - 1 = 2^{n-1} + 2^{n-2} + \dots + 2^2 + 2^1 + 2^0.$$

Но оно не показалось бы очевидным, если, не зная заранее ответа, пришлось бы вычислять сумму  $1 + 2 + 4 + 8 + \dots + 2^{n-2} + 2^{n-1}$ , называемую суммой геометрической прогрессии со знаменателем 2.

Ведь для этого пришлось бы выдумать какой-нибудь трюк наподобие следующего:

$$\begin{aligned} 1 + 2 + 4 + 8 + \dots + 2^{n-2} + 2^{n-1} &= \\ = 2 - 1 + 2 + 4 + 8 + \dots + 2^{n-2} + 2^{n-1} &= \\ = 4 - 1 + 4 + 8 + \dots + 2^{n-2} + 2^{n-1} &= \\ = 8 - 1 + 8 + 16 + \dots + 2^{n-2} + 2^{n-1} &= \dots = \\ = 2^{n-2} - 1 + 2^{n-2} + 2^{n-1} = 2^{n-1} - 1 + 2^{n-1} &= 2^n - 1. \end{aligned}$$

2. Используя подобный трюк, вычислите произведение  $(2+1)(2^2+1)\dots(2^{2^n}+1)$ .

Докажем теперь существование и единственность представления числа  $N$  в виде суммы меньших степеней двойки. Доказательство будем проводить индукцией по  $N$ .

Для  $N = 1$  утверждение очевидно.

Пусть оно верно для всех  $N \leq N_0$ . Пусть  $2^n$  — максимальная степень двойки, не превосходящая  $N$ , т.е.  $2^n \leq N_0 < 2^{n+1}$ . Тогда по предположению индукции число  $N_0 - 2^n \leq 2^n$  представимо в виде суммы степеней двойки, меньших  $N_0 - 2^n < 2^n$ . Следовательно, число  $N_0$  тоже представимо в виде суммы меньших степеней двойки (достаточно к представлению числа  $N_0 - 2^n$  добавить  $2^n$ ). Кроме того, так как  $1 + 2 + \dots + 2^{n-1} = 2^n - 1 < 2^n$ , то не существует представления числа  $N_0$ , не использующего  $2^n$ . Таким образом, доказана единственность такого представления.

Заметим, что для быстрого применения этого алгоритма удобно заранее вычислить все степени двойки, не превосходящие данного числа.

Заметим еще, что, в отличие от первого варианта решения, полная сумма во всех конвертах менее чем в два раза превосходит верхнюю границу подлежащей выплате суммы.

\* Печатается по тексту одноименной книги. Издательство МЦНМО, 2004.

<sup>1</sup> Двумя чертами слева выделены тексты задач для самостоятельного решения.

Для краткой записи результата работы алгоритма над данным числом  $a$  можно вместо разложения  $a = 2^{n_1} + \dots + 2^{n_k}$ , которое и записать-то в общем виде без использования трехэтажных обозначений затруднительно, использовать последовательность показателей степеней  $(n_1, \dots, n_k)$ , или, что еще удобнее (но не всегда короче), написать последовательность  $(a_m, \dots, a_1)$  чисел 0 и 1, в которой  $a_i = 1$ , если число  $2^{i-1}$  входит в указанное выше разложение, и  $a_i = 0$  в противном случае. Тогда это разложение можно будет переписать в виде  $a = a_1 + 2a_2 + 4a_3 + \dots + 2^{m-1}a_m$ .

Ясно, что приведенный выше алгоритм позволяет строить такое представление, причем оно определяется однозначно, если предполагать, что старший его разряд  $a_m$  ненулевой. Это представление и называется двоичной записью числа  $a$ .

Читатель увидит, что понятие двоичной записи очень похоже на понятие десятичной записи и в каком-то смысле даже проще.

Остался вопрос о минимальности найденной системы конвертов. В общем виде указанный выше прием предлагает для уплаты любой суммы от 1 до  $n$  использовать  $m$  конвертов с суммами 1, 2, 4, 8, ...,  $2^{m-1}$ , где  $2^{m-1} \leq n < 2^m$ . Меньшего количества конвертов может не хватить, потому что с помощью  $k < m$  конвертов можно уплатить не более чем  $2^k - 1 < 2^{m-1} \leq n$  разных сумм, так как каждая сумма однозначно определяется ненулевым набором  $(a_1, \dots, a_k)$ , в котором каждое число  $a_i$  равно 1, если  $i$ -й конверт входит в эту сумму, и равно 0 в противном случае, а всего наборов длины  $k$  из нулей и единиц можно составить ровно  $2^k$ .

3. Докажите последнее утверждение.

4. Докажите, что если  $n = 2^m - 1$ , то минимальная система конвертов определяется однозначно, в противном случае — нет.

После упоминания десятичной системы сразу возникает идея на первый взгляд даже более простого решения задачи о конвертах. Надо просто заготовить конверты с суммами 1, 2, ..., 9, 10, 20, 30, ..., 90, 100, 200, 300, ..., 100 000 000, 200 000 000, ..., 900 000 000. Тогда для выплаты любой требуемой суммы не нужно искать ее двоичную запись, так как для выплаты, например, 123 456 789 у. е. нужно просто взять конверт с суммой 9, конверт с суммой 80, конверт с суммой 700 и т.д. Это действительно проще, но исключительно потому, что мы привыкли пользоваться десятичной системой и все расчеты ведутся с ее помощью. Если бы мы использовали в повседневной жизни только двоичную систему, то этот способ был бы сложнее, так как приходилось бы переводить данную сумму из двоичной системы в десятичную<sup>2</sup>. Поэтому простота десятичного способа решения задачи, скорее, мнимая.

На самом деле указанный выше двоичный метод имеет преимущество перед десятичным (и любым другим). Оно заключается в меньшем числе используемых конвертов, что было показано выше. Хотя длина двоичной записи числа в три с лишним раза больше длины его десятич-

ной записи, на каждую цифру десятичной записи приходится девять конвертов, т.е. число конвертов в двоичном методе почти в три раза меньше, чем в десятичном.

Идея, лежащая в основе изложенной задачи, видимо, очень древняя, и происходит, вероятно, из Индии. Об этом свидетельствует легенда об изобретателе шахмат, который скромно попросил (после настояний магараджи, которому очень понравилась игра) себе в награду положить одно зерно на угловую клетку шахматной доски и удваивать количество зерен на каждой следующей клетке. Магараджа, подивившись скудоумию казавшегося таким мудрым человека, распорядился отсыпать ему запрошенные несколько мешков зерна.

5. Оцените приблизительно, во сколько миллионов тонн зерна обойдется магарадже его щедрость.

Из сказанного выше видно, что если бы на каждое поле шахматной доски не всегда класть столько зерна, сколько просил мудрец, а иногда вообще не класть зерен, то можно получить таким образом любое число от 0 до  $2^{64} - 1$ . Поэтому, вероятно, таким образом можно представить любое число, которое может встретиться в каких-либо конкретных прикладных вычислениях.

Индийская легенда обращает наше внимание на одну особенность двоичной (и любой позиционной) системы — возможность представить колоссальные числа в виде короткой записи. Разумеется, в качестве такой записи не надо использовать совокупность количеств зерен, лежащих на клетках доски в точности так, как указано выше, — ведь эти числа могут быть очень велики, и реально такое количество зерен на большей части клеток доски поместиться не может. Вместо этого, как и принято в двоичной системе, на каждую клетку или не кладется зерен вообще, или кладется одно зерно, которое символизирует соответствующую степень двойки. Тогда шахматная доска превращается, по существу, в то, что на Востоке называют абак, а в России — счеты.

Конечно, реально используемые счеты всегда были десятичными, но проведенные выше рассуждения показывают, что, хотя двоичная запись в три раза длиннее десятичной (и вообще, из всех позиционных систем в этом смысле двоичная — самая плохая), но изготовление счет с применением двоичной системы могло бы дать определенную (правда, лишь теоретическую) экономию.

6. Пусть на каждой из  $n$  спиц счет находится по  $b$  костяшек (т.е. счеты представляют числа в системе счисления с основанием  $b + 1$ ), и поэтому они позволяют записать в этой системе любое число от 0 до  $N = (b + 1)^n - 1$  (число  $N$  характеризует «вместимость» счетов). Каким нужно выбрать  $b$ , чтобы суммарное количество костяшек на счетах («сложность» счет) было минимальным при условии возможности указанного представления на счетах любого числа от 1 до  $N$  (т.е. при заданной вместимости)?

Для прочитавших этот параграф ответ, конечно, очевиден. Для знающих логарифмы продолжение этой задачи: сравните сложности десятичных и двоичных счет одинаковой вместимости.

<sup>2</sup> Иногда это приходится делать и в реальной жизни. Различные алгоритмы такого перевода будут изложены далее.

Приведенный выше алгоритм перевода из десятичной системы в двоичную вычислял цифры двоичной записи, начиная со старших цифр. Опишем теперь кратко возможно более удобный алгоритм, в котором цифры двоичной записи вычисляются, начиная с младших<sup>3</sup>.

Очевидно, самая младшая цифра равна нулю, если число четное, и единице, если оно нечетное. Для нахождения остальных двоичных цифр надо от исходного числа отнять найденную младшую цифру, поделить разность пополам и к полученному числу применить описанный выше шаг алгоритма.

Например, у числа 300 последние две цифры нули, а для нахождения остальных цифр надо иметь дело с числом  $300/4 = 75$ , поэтому следующая цифра 1, и получаем промежуточный результат 37.

Следующая далее цифра опять 1, и промежуточный результат 18, поэтому следующая цифра 0, а промежуточный результат 9, следующая цифра 1, а потом три нуля подряд, а старший разряд, как всегда, 1. В результате получается двоичная запись 1000101100.

Преимущество этого алгоритма в том, что не требуется предварительного вычисления степеней двойки, но зато приходится неоднократно выполнять операцию деления пополам.

## § 2. Взвешивание с помощью гирь и возведение в степень

Предлагаем читателю самому убедиться в том, что точно так же, как и в предыдущем разделе, можно доказать, что для отвешивания любого числа граммов песка от 1 г до  $n$  г за одно взвешивание достаточно иметь гири 1 г, 2 г, 4 г, ...,  $2^m$  г, где  $2^m \leq n < 2^{m+1}$ , и меньшего числа гирь недостаточно, если песок лежит на одной чашке весов, а гири разрешается ставить на вторую чашку. На самом деле с математической точки зрения эта задача, известная со средневековых времен, ничем не отличается от рассмотренной выше задачи о конвертах с деньгами.

Часто новые и интересные задачи получаются, если в старой задаче наложить какие-нибудь естественные ограничения. Например, можно задать следующий вопрос: за какое наименьшее количество взвешиваний на чашечных весах можно отвесить килограмм сахарного песка, если имеется лишь одна однограммовая гирька?

На первый взгляд кажется, что единственный способ решения этой задачи — отвесить один грамм, положить в эту же чашку гирьку, отвесить в другой чашке два грамма, переложить гирьку в нее и т.д., добавляя по одному грамму, после тысячного взвешивания отмерить наконец-то килограмм.

Но есть и более быстрый способ. Нужно лишь заметить, что если мы научились отвешивать за  $n$  взвешиваний  $m$  г песка, то, сделав еще одно взвешивание, можно, даже не используя гирьку, отвесить еще  $m$  г и, сыпав обе порции вместе, получить  $2m$  г за  $n + 1$  взвешивание.

А если при этом взвешивании положить на одну из чашек гирьку, то за  $n + 1$  взвешивание можно отмерить  $2m \pm 1$  г песка.

Теперь воспользуемся двоичной записью числа 1000. Применяя любой из указанных выше алгоритмов, получаем равенство  $1000 = 2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^3$ .

$$\begin{aligned} & \text{Так как } 2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^3 = \\ & = (((((2 + 1)2 + 1)2 + 1)2 + 1)2^2 + 1)2^3, \end{aligned}$$

то, последовательно отвешивая 1,  $2 + 1 = 3$ ,  $2 \cdot 3 + 1 = 7$ ,  $2 \cdot 7 + 1 = 15$ ,  $2 \cdot 15 + 1 = 31$ ,  $2 \cdot 31 = 62$ ,  $2 \cdot 62 + 1 = 125$ ,  $2 \cdot 125 = 250$ ,  $2 \cdot 250 = 500$ , получаем на десятом взвешивании  $2 \cdot 500 = 1000$  г. Девяти взвешиваний не хватит, потому что за два взвешивания можно отмерить массу не более  $3 = 2^2 - 1$ , за три — не более  $7 = 2 \cdot 3 + 1 = 2^3 - 1$ , за четыре — не более  $15 = 2 \cdot 7 + 1 = 2^4 - 1$ , и за девять взвешиваний — не более  $511 = 2^9 - 1$ .

Если нужно отмерить  $n$  г песка, то надо записать  $n$  в двоичном виде  $a_m \dots a_1$ , где  $2^{m-1} \leq n < 2^m$ ,  $a_m = 1$ , и воспользоваться формулой  $n = a_m 2^{m-1} + \dots + a_2 2 + a_1 = (\dots((2a_m + a_{m-1})2 + a_{m-2})\dots)2 + a_1$ , последовательно отвешивая по  $b_1 = a_m$ ,  $b_2 = 2b_1 + a_{m-1}$ ,  $b_3 = 2b_2 + a_{m-2}$ , ...,  $b_m = b_{m-1}2 + a_1 = n$  г.

В используемой формуле знающие читатели увидят так называемую “схему Горнера”. К ней мы еще вернемся в дальнейшем.

Идея, лежащая в основе этого метода взвешивания, стара как сама математика. Ее применяли и древние египтяне, и древние индусы, но, конечно, не для взвешивания, а для умножения. Ведь алгоритм умножения столбиком был придуман не сразу, а до этого умножение сводилось к сложению. Например, чтобы умножить какое-нибудь число  $a$  на 1000, можно, используя только операции сложения, последовательно вычислить  $a + a + a = 3a$ ,  $3a + 3a + a = 7a$ ,  $7a + 7a + a = 15a$ ,  $15a + 15a + a = 31a$ ,  $31a + 31a = 62a$ ,  $62a + 62a + a = 125a$ ,  $125a + 125a = 250a$ ,  $250a + 250a = 500a$ ,  $500a + 500a = 1000a$ . Такой метод умножения дожил почти до нашего времени, он удобен при вычислениях на счетах. Сейчас он никому не нужен, так как все используют калькуляторы. Но как возвести на калькуляторе число  $a$ , например, в тысячную степень, если у него нет специальной операции возведения в произвольную степень? Умножать 999 раз не нужно, а можно применить тот же прием, последовательно вычисляя  $a^3 = a^2 a$ ,  $a^7 = (a^3)^2 a$ ,  $a^{15} = (a^7)^2 a$ ,  $a^{31} = (a^{15})^2 a$ ,  $a^{62} = (a^{31})^2$ ,  $a^{125} = (a^{62})^2 a$ ,  $a^{250} = (a^{125})^2$ ,  $a^{500} = (a^{250})^2$ ,  $a^{1000} = (a^{500})^2$ .

Если вспомнить, что 1000 имеет двоичную запись 111101000, то можно заметить, что если отбросить старший бит (всегда равный единице), то каждому следующему биту соответствует операция возведения в квадрат, если он нулевой, или, если он ненулевой, возведение в квадрат с последующим умножением на число  $a$  — основание степени (т.е. делается две операции). Кстати, число  $a$  не нужно каждый раз заново набирать на клавиатуре. Нужно в самом начале вычислений занести его в память калькулятора, и когда нужно, после нажатия кнопки для

<sup>3</sup> Кстати, кассиры в магазинах и на рынках предпочитают выдавать сдачу начиная с мелких купюр, вопреки инструкции. Причина понятная — надеются, что покупатель, получив мелочь, уйдет, забыв взять крупные.

умножения, просто вызывать его из памяти и потом нажать кнопку “равно”. Таким образом, возведение в квадрат требует двукратного нажатия кнопок, а возведение в квадрат и последующее умножение на основание степени — пятикратного. Для того чтобы не запутаться в операциях, можно перед началом вычислений составить mnemonic-правило. Возведение в квадрат обозначим символом К, а возведение в квадрат и последующее умножение — символом КУ. Тогда, заменяя в двоичной записи единицы (кроме старшей) на КУ, а нули — на К, получим правило КУКУКУКУКУКУКК, или короче  $(КУ)^4ККУК^3$ .

Посчитаем общее число операций умножения в рассмотренном вычислении. Число возведений в квадрат на единицу меньше длины двоичной записи показателя степени, а число умножений общего вида на единицу меньше суммы цифр двоичной записи.

Для любого  $n$  обозначим  $\lambda(n)$  уменьшенную на единицу длину двоичной записи числа  $n$ , а  $v(n)$  — ее сумму цифр (другими словами, число единиц в ней). Тогда в общем случае число операций умножения, использованных в этом методе возведения в степень  $n$ , будет равно  $\lambda(n) + v(n) - 1$ . Далее будет показано, что меньшим числом операций обойтись нельзя, если только не обновлять содержимое ячейки памяти.

Очевидно, что  $\lambda(n) + v(n) - 1 \leq 2\lambda(n)$ . Те, кто знают логарифмы, сообразят, что  $\lambda(n) = \lfloor \log_2 n \rfloor$ , где знак  $\lfloor x \rfloor$  означает целую часть числа  $x$ . Но можно вычислить обе введенные функции даже не упоминая о двоичной записи. Для этого надо воспользоваться следующими правилами:  
 $v(1) = 1, v(2n) = v(n), v(2n + 1) = v(n) + 1,$   
 $\lambda(1) = 0, \lambda(2n) = \lambda(2n + 1) = \lambda(n) + 1.$

Однако для доказательности справедливости этих правил полезно, конечно, воспользоваться двоичной системой, после чего они становятся почти очевидными.

Докажем полезное и простое неравенство  $v(n + 1) \leq v(n) + 1$ . Оно очевидно превращается в равенство, если  $n$  четно, так как тогда его двоичная запись заканчивается нулем. Если же эта двоичная запись заканчивается  $k$  единицами, перед которыми стоит ноль, то двоичная запись числа  $n + 1$  заканчивается  $k$  нулями, перед которыми стоит единица (а старшие биты остаются без изменения, если они есть). Для того чтобы в этом убедиться, достаточно выполнить прибавление 1 к  $n$  в двоичной системе. В обоих рассмотренных случаях  $v(n + 1) \leq v(n) + 1$ .

Из доказанного неравенства следует, что  $\lambda(n + 1) + v(n + 1) \leq \lambda(n) + v(n) + 1$ . Действительно, если  $2^{k-1} < n + 1 < 2^k$ , то  $\lambda(n + 1) = k - 1 = \lambda(n)$ , и из неравенства  $v(n + 1) \leq v(n) + 1$  следует нужная нам оценка. Если же  $n + 1 = 2^k$ , то  $\lambda(n + 1) = k = \lambda(n) + 1, v(n + 1) = 1, v(n) = k$ , откуда следует, что  $\lambda(n + 1) + v(n + 1) = k + 1 \leq 2k = \lambda(n) + v(n) + 1$ .

Справедливо также равенство  $\lambda(2n) + v(2n) = \lambda(n) + v(n) + 1$ , которое сразу следует из равенств  $v(2n) = v(n), \lambda(2n) = \lambda(n) + 1$ .

Выше было показано, что число операций умножения, использованных для возведения в степень  $n$  на калькуляторе с одной ячейкой памяти, не больше чем  $\lambda(n) + v(n) - 1$ . При  $n = 1, 2$  очевидно, что меньшим числом операций обойтись нельзя. Покажем, что и в

общем случае это так, если только не обновлять содержимое ячейки памяти, т.е., кроме возведения в квадрат, всегда использовать только умножение на основание степени.

Допустим противное, а именно, что для вычисления указанным образом  $x^n$  при некотором  $n$  оказалось достаточно  $l < \lambda(n) + v(n) - 1$  операций. Выберем среди таких чисел  $n$  наименьшее число и обозначим его также  $n$ . Если последняя операция в рассматриваемом вычислении была возведение в квадрат, то  $n$  четно, и для вычисления  $x^{n/2}$  достаточно  $l - 1 < \lambda(n) + v(n) - 2 = \lambda(n/2) + v(n/2) - 1$  операций, поэтому минимальным числом с рассматриваемым свойством не может быть  $n$ , что ведет к противоречию.

Аналогично получается противоречие и в случае, когда последней операцией было умножение на  $x$ . Действительно, тогда согласно доказанному выше неравенству для вычисления  $x^{n-1}$  достаточно  $l - 1 < \lambda(n) + v(n) - 2 \leq \lambda(n - 1) + v(n - 1) - 1$  операций.

Но если обновлять содержимое ячейки памяти, то указанный выше метод вычисления  $x^{1000}$  можно улучшить. Для этого можно применить так называемый “метод множителей”. Идея этого метода заключается в следующем. Заметим, что если мы умеем возводить в степень  $n$  за  $l(n)$  операций и возводить в степень  $m$  за  $l(m)$  операций, то можно после того, как закончено вычисление  $x^n$ , занести его в ячейку памяти и далее вычислить  $x^{nm} = (x^n)^m$  за  $l(m)$  операций, используя тот же метод, что и для вычисления  $x^n$ . Тогда общее число операций будет равно  $l(nm) = l(n) + l(m)$ .

Вычисляя  $x^5$  старым методом за  $\lambda(5) + v(5) - 1 = 3$  операции (с помощью последовательности  $x, x^2, x^4 = (x^2)^2, x^5 = (x^4)x$ ) и применяя два раза метод множителей, получаем, что  $l(125) = 3l(5) = 9$ . Выполняя еще три возведения в квадрат, получаем  $l(1000) = l(125) + 3 = 12$ . Старый же метод требовал  $\lambda(1000) + v(1000) - 1 = 9 + 6 - 1 = 14$  операций.

Читателю может показаться, что мы слишком много внимания уделили такому специальному и не слишком важному вопросу, как быстрое выполнение возведения в степень. Лет тридцать назад это замечание было бы справедливым. Но в середине 1970-х годов почти одновременно и независимо группой американских математиков (У.Диффи, М.Хеллман, Р.Ривест, А.Шамир, П.Адлеман) и группой английских криптографов (К.Кокс, М.Вильямсон, Д.Эллис) были открыты первые алгоритмы криптографии с открытым ключом<sup>4</sup>. Благодаря этим алгоритмам теперь частные лица могут обмениваться секретной информацией по общедоступным каналам связи (например, по Интернету) без боязни, что их сообщения прочтут не только конкуренты, но и спецслужбы. Возникшее направление в криптографии быстро превратилось в популярную область математических исследований, которой уже посвящены многочисленные журналы и книги. И во многих самых распространенных алгоритмах важную роль играет операция возведения в степень.

<sup>4</sup> Англичане сделали это раньше, но им, как сотрудникам секретной криптографической службы, было запрещено опубликовать свои результаты в открытой печати.

### § 3. Аддитивные цепочки и флаги с молоком

Назовем *аддитивной цепочкой* любую начинающуюся с 1 последовательность натуральных чисел  $a_0 = 1, a_1, \dots, a_m$ , в которой каждое число является суммой каких-то двух предыдущих чисел (или удвоением какого-то предыдущего числа). Обозначим  $l(n)$  наименьшую длину аддитивной цепочки, заканчивающейся числом  $n$  (длиной цепочки  $a_0 = 1, a_1, \dots, a_m$  называем число  $m$ ).

Например, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 — аддитивная цепочка, 1, 2, 3, 5, 7, 14 — минимальная цепочка для 14, т.е.  $l(14) = 5$ . Аддитивные цепочки можно изображать в виде ориентированного графа, в котором в вершину  $a_i$  идут ребра от вершин  $a_j, a_k$ , если  $a_i = a_j + a_k$  (в случае, если такое представление неоднозначно, выбираем любое из них и рисуем только два ребра). Если из какой-то вершины выходит только одно ребро, то для краткости можно “склеить” эту вершину с той вершиной, в которую ведет это ребро. Граф для предыдущего примера показан на рис. 1.

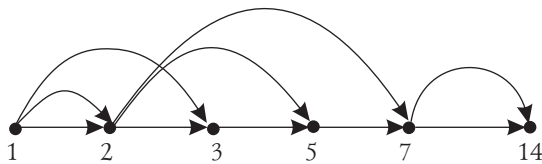


Рис. 1

Можно считать, что все числа в цепочке разные, так как этого легко достичь просто удаляя из нее повторяющиеся числа, и что эти числа расположены в цепочке в порядке возрастания.

Очевидно, что наименьшее число умножений, необходимое для возведения в  $n$ -ю степень, равно  $l(n)$ .

Приведенный выше метод построения аддитивных цепочек называется двоичным (или бинарным). Фактически этим методом было доказано, что справедливо неравенство  $l(n) \leq \lambda(n) + v(n) - 1$ . Методом множителей легко доказать неравенство  $l(nm) \leq l(n) + l(m)$ .

7. Докажите нижнюю оценку:  $l(n) \geq \lambda(n)$ .

Из этой оценки следует, что  $l(2^n) = n$ .

Интересно, что бинарный метод был, по существу, известен древним индусам, потом был переоткрыт арабскими математиками, задача о точном вычислении функции  $l(n)$  появилась в одном французском журнале в 1894 году, потом заново была переоткрыта в 1930-е годы и неоднократно ставилась в дальнейшем, но до сих пор в общем случае не решена.

По существу, наилучшая из известных общих верхних оценок была доказана в 1930-е годы А.Брауэром и имеет вид:

$$l(n) \leq \lambda(n) \left( 1 + \frac{1}{\lambda(\lambda(n))} + \frac{C\lambda(\lambda(\lambda(n)))}{(\lambda(\lambda(n)))^2} \right),$$

— где  $C > 0$  — некоторая константа.

Не каждую аддитивную цепочку можно вычислить на калькуляторе с одной ячейкой памяти, не используя для запоминания промежуточных результатов собственную голову (фактически такие калькуляторы имеют две ячейки памяти, так одна из них содержит число, изображае-

мое в данный момент на дисплее). Укажем, как можно определить необходимое число ячеек памяти для вычисления данной аддитивной цепочки. Для этого введем понятия ширины (а заодно и глубины) аддитивной цепочки.

Пусть дана произвольная цепочка  $a_0 = 1, a_1, \dots, a_l = n$ . Сопоставим каждому ее элементу два числа. Первое из них назовем *глубиной элемента*, а второе — *номером ячейки*, хранящей это число. Для элемента  $a_0$  первое число положим равным нулю, а второе — единице. Будем далее последовательно вычислять эти числа для элементов цепочки. Пусть они уже вычислены для всех элементов от  $a_0$  до  $a_k$ . Составим список номеров ячеек, содержащих те элементы цепочки, которые еще могут быть использованы для вычисления последующих элементов. Найдем наименьшее число, не входящее в этот список, и присвоим его элементу  $a_{k+1}$  в качестве номера ячейки (возможно, она использовалась ранее, но теперь уже свободна). Пусть  $a_{k+1} = a_i + a_j$ ,  $i, j \leq k$ . Если  $D(a_i), D(a_j)$  — значения глубины элементов  $a_i, a_j$ , то положим  $D(a_{k+1})$  на единицу большим максимального из чисел  $D(a_i), D(a_j)$ . *Шириной  $S$*  цепочки назовем число использованных ячеек (равное наибольшему из использованных номеров ячеек). *Глубиной  $D$*  цепочки назовем глубину ее последнего элемента.

8. Докажите, что  $a_k \leq 2^{D(a_k)}$  и  $D \geq \lambda(n)$ . Докажите, что бинарный метод можно модифицировать так, чтобы длина цепочки не изменилась, а глубина стала бы равна  $\lambda(n)$ .

Если цепочка имеет ширину  $S$ , то ее можно представить в виде вычисления на калькуляторе с  $S - 1$  ячейками памяти (кроме основной, содержащей число, изображаемое в данный момент на дисплее) или в виде компьютерной программы, использующей  $S$  ячеек памяти.

Можно еще представить эту цепочку в виде способа, как налить в данную флягу  $n$  литров молока из цистерны, если первоначально в ней был один литр и, кроме нее, имеется  $S$  таких же пустых фляг и весы, способные только сравнивать веса двух фляг между собой.

Для этого сопоставим  $S$  фляг ячейкам памяти рассматриваемой цепочки, а одну флягу оставим запасной. Тогда любую операцию с ячейками памяти вида  $x_k = x_i + x_j$  можно выполнить, выливая в случае необходимости  $k$ -ю флягу в цистерну, потом наливая запасную флягу до уровня  $i$ -й фляги и сливая ее содержимое в  $k$ -ю флягу, если  $k \neq i$ , и делая аналогичную процедуру для индекса  $j$ .

Естественно, что аналогичным образом на языке “переливаний” можно представить и программу с командами, использующими не только сложение, но и вычитание  $x_k = x_i - x_j$ . Поэтому понятие аддитивной цепочки можно обобщить, разрешив использовать вычитание.

Для вычисления степеней такие цепочки также можно выполнять на калькуляторе, если, кроме умножения, использовать и деление. Известно, что в среднем это не дает существенной выгоды, но в некоторых случаях число используемых операций уменьшается.

Например, вычислить  $x^{1000}$  можно с помощью следующей цепочки: 1, 2, 4, 8, 16, 32, 31, 62, 124, 125, 250, 500, 1000.

Продолжение в следующем номере

# НАЧАЛКА

газета-клуб для всех,  
кто учит информатике  
маленьких детей



№ 2 (16–31 января)

## ”Мир информатики” для младших школьников

А.В. МОГИЛЕВ,  
г. Воронеж

### 1. Зачем и чему учить младших школьников в информатике?

Вашему вниманию предлагаются размышления об использовании в учебном процессе начальной школы учебно-методического комплекса “Мир информатики” для первого, второго и третьего годов обучения, вышедшего в 2003–2005 годах в издательстве “Ассоциация XXI век” [1–9].

Кто должен вести курс начальной информационной подготовки в школе? Было бы неправильным адресовать это пособие, скажем, только учителям начальной школы или только преподавателям информатики. Во время дискуссий на эту тему в пользу той или иной позиции обычно выдвигаются аргументы типа “учителя начальных классов не обладают необходимой компьютерной подготовкой” или “учителя информатики не имеют достаточного опыта работы с детьми младшего школьного возраста”. Все эти аргументы довольно спорны, и можно найти довольно весомые доводы как в пользу, так и против той и другой точки зрения. По нашему (авторов представляемого цикла пособий) мнению, в текущей ситуации в российском образовании, когда только начинается обновление образования в начальной школе и идет поиск места информационной подготовки в ней, вести новый предмет могут те, кто к этому готов, то есть обладает не только необходимой подготовкой (и компьютерной, и методической в плане работы со школьниками младшего возраста), но и желанием.

К сожалению, долгое время начальная школа оставалась наиболее консервативной частью общего образования. Содержание обучения в начальной школе сложилось в конце 20-х — начале 30-х годов XX века. С тех пор его изменения, да и сам подход к обучению младших школьников, формирующий в основном исполнительские компетенции, были незначительны. Выполнение большого числа стереотипных упражнений, доминирование учителя в ходе урока, акцент на так называемой “дисциплине

на уроке” — вот неполный перечень примет российского начального образования, доставшийся нам от трудной эпохи в нашей истории. Современное состояние общества ставит перед образованием, особенно начальным, категорический императив — перейти от подготовки простых исполнителей, способных действовать по образцу, в соответствии с правилами, к формированию свободной, инициативной и творческой личности каждого ребенка. Слово “формирование” в этом предложении — также из прошлого века. Личность формируется в результате саморазвития, задача начальной школы — создать для этого рациональные, благоприятные условия.

Одним из направлений диверсификации начального образования является включение в него информационной подготовки. На протяжении последних 4–5 лет оно протекает с переменным успехом, в упорной борьбе носителей “охранительных” взглядов, прикрывающихся подчас лозунгами заботы о здоровье детей, и новаторов, иногда отрывающихся от реальной школьной почвы, переходящих за грань возможного. Предмет “информатика” вот уже около 20 лет с момента ее введения в предметную систему школы остается наиболее быстроменяющимся и революционизирующим школьное образование предметом. Впрочем, совершенно такую же роль информатика играет и в масштабах всего общества.

**Информатика** в настоящее время — развитая наукоемкая сфера деятельности, связанная с передачей, хранением, преобразованием и использованием информации с помощью компьютерных систем, реализующая системно-информационный подход к познанию окружающего мира, имеющая тенденцию к превращению в фундаментальную отрасль научного знания об информационных процессах в природе и обществе.

Информационная подготовка — одно из немногих инновационных и востребованных направлений школьной подготовки, делающих школу современной, прибли-

жающих ее к жизни и запросам общества. В актуальных государственных программах развития образования России информатизация рассматривается как важнейший аспект модернизации образования.

Средства информатизации, терминология, связанная с информационными процессами и системами, способы деятельности в информационной среде обступают учащихся школы с самого раннего детства, становятся для них естественным окружением. Общественные запросы ставят перед образованием задачу формирования информационной культуры учащихся. Информационная культура становится ведущей составляющей подготовки человека, в какой бы сфере деятельности ему ни пришлось работать в будущем.

Большинство целей информационной подготовки, отражаемых в научно-методической литературе, специфичны для основной школы и старших классов. Среди них

- *формирование информационного мировоззрения: информационной картины мира, общности закономерностей информационных процессов в системах различной природы; ценностного отношения к информации;*
- *формирование представлений о роли и месте информационных технологий, информационном содержании трудовых процессов в обществе;*
- *выработка стабильных навыков получения и обработки ориентированной на индивидуальные личностные запросы информации;*
- *развитие способностей к адаптации в изменяющейся информационной среде деятельности;*
- *пропедевтика дальнейшей информационной подготовки в течение всей жизни.*

В начальной школе эти цели, проецируясь на возрастные потребности и возможности детей, определяют основные задачи курса «Мир информатики»:

- *формирование представлений об информационной природе процессов в живой природе, обществе и технике, ценностного отношения к информации;*
- *знакомство с основными компонентами компьютера и их функциями; формирование представлений о типах и назначении компьютерных программ;*
- *освоение способов деятельности в информационной среде, умений решать разнообразные задачи с помощью компьютера, программ, алгоритмов; формирование навыков их рационального использования для удовлетворения собственных информационных потребностей;*
- *интеллектуальное развитие, развитие когнитивных и творческих способностей и навыков учащихся в процессе работы с развивающими, игровыми и другими программами;*
- *расширение кругозора детей, закрепление и совершенствование знаний, полученных в других курсах начальной школы;*
- *подготовка к дальнейшему обучению информатике в основной школе.*

Для решения этих образовательных задач в первую очередь необходимо провести рациональный отбор и структурирование содержания обучения, в соответствии с ними определить применяемые методы обучения. Учебные пособия в определенном смысле являются сплавом содержания образования и применяемых методов обучения.

### Принципы отбора содержания обучения для настоящего курса

Как обычно отбирается содержание обучения для школьного предмета? Как правило, это длительный процесс, происходящий если не в течение столетий, то по крайней мере десятилетий. Оно определяется множеством явно и неявно действующих факторов. Среди них мы можем выделить

- 1) состояние соответствующей научной области или практики производственно-хозяйственной деятельности;
- 2) потребности общества в том, чтобы его члены обладали определенными знаниями, навыками, находились на определенных мировоззренческих и нравственных позициях;
- 3) потребности личности учащегося прежде всего в самореализации, в развитии интеллектуальной, когнитивной, мотивационной сфер;
- 4) системные связи между компонентами самого образования, потребности образовательной сферы, традиции, консерватизм в подготовке учительских кадров;
- 5) традиции или требования, поддерживаемые той или иной из социальных групп или даже влиятельных личностей;
- 6) соображения обеспеченности средствами обучения определенному содержанию обучения в образовательных учреждениях;
- 7) процесс постоянного и «самопроизвольного» усложнения, углубления содержания образования от учебной программы к программе, от учебника к учебнику (ведь надо же как-то обосновывать смену программ и учебных пособий!).

Эти одновременно, с разной силой и противоречиво действующие факторы приводят к тому, что бывает очень трудно установить причины наличия тех или иных тем в образовательных стандартах и учебниках, если речь идет о стабильном, давно существующем учебном предмете. Однако в отношении информатики как учебного предмета, который только переживает свое становление, мы сталкиваемся с действием меньшего числа факторов, а именно факторов 1–5, причем действие их достаточно явно и поддается анализу.

Применительно к отбору содержания обучения информатике в младшей школе, по нашему мнению, следует придавать наибольший вес третьему фактору, потребностям личности учащегося в развитии интеллектуальной, когнитивной, мотивационной сфер. В современной системе образования в младшей школе делается упор на формирование так называемых «учебных» умений и навыков, при этом развивающий компонент слабо представлен в учебной деятельности. Мы настаиваем на том, что в информатике в младшей школе развитие ребенка должно иметь приоритет. Это не отменяет, однако, и необходимости формирования общеучебных навыков, поддержки обучения другим предметам — это проявление фактора 4). Также нельзя пренебречь фактором 2) — формированием у учащихся представлений об информационной картине мира, ценностного отношения к информации, равно как и фактором 1) — отражением современного состояния научно-практической сферы, связанной с информатикой (он соответствует дидактическому принципу научности).



При отборе и структурировании содержания обучения необходимо учитывать и ряд требований дидактического порядка, в первую очередь принципы системности и доступности. В соответствии с этими принципами следует выделять наименьшее возможное количество модулей (дидактических единиц) содержания обучения.

Детальный анализ указанных факторов и требований, а также требование доступности — простоты приводит к тому, что в содержании информационной подготовки в младшей школе следует выделить всего три содержательных модуля — дидактических единицы:

I. Информация и информационные процессы (и основы алгоритмизации).

II. Компьютер и его периферийные устройства. Программные средства.

III. Информационные и коммуникационные технологии.

Содержание этих модулей достаточно обособлено друг от друга, учебная деятельность при освоении этих модулей по своему характеру специфична, вести разработку материала, планирование и организацию учебного процесса можно независимо. В некоторых образовательных программах можно столкнуться с большим количеством содержательных блоков или “линий”, однако провести границы между такими “линиями” можно лишь формально, учебная деятельность не обладает специфическими отличиями при их освоении, а неоправданное дробление и детализация дидактических единиц не способствуют рациональному планированию и организации учебного процесса, а также формированию системы компетенций учащихся.

Отбор блока “Информация и информационные процессы” обусловлен фактором 2) отбора содержания обучения — требованием закладки в младшем школьном возрасте основ мировоззрения. Этот блок часто называют в пособиях для старших классов “Теоретической информатикой”. Это неверно. Во-первых, такого раздела науки не существует. Это все равно, что говорить “теоретическая математика” или “теоретический русский язык”. Во-вторых, такое название подчеркивает, что обучение будет носить теоретико-ориентированный характер, то есть, как всегда, оторвется от практической деятельности, от текущих запросов и потребностей как формирующейся личности самого ребенка, так и общества в целом, и в этом обучении будет доминировать учитель как основной источник информации.

Кроме того, учебная деятельность этого модуля в начальной школе будет иметь явный развивающий характер. Важно, что знакомство с психическими процессами восприятия информации с помощью органов чувств, ее запоминание, переработка в процессе мышления с точки зрения информатики помогает учащимся осознанно управлять собственной познавательной деятельностью в процессе обучения.

Блок II — “Компьютер и его устройства. Программные средства” обусловлен учетом 1-го и 4-го факторов, он отражает современное состояние средств компьютеризации — аппаратного и программного обеспечения. Важным, связующим звеном всего материала является содержательный блок III — он определяется факторами 1, 3 и 4, т.е. одновременно знакомит учащихся с современными применениями средств информатизации, обеспечивает разви-

тие интеллектуальной сферы ребенка и одновременно формирует общеучебные навыки, в том числе связанные с использованием компьютера в учебной деятельности.

Отобрав материал, необходимо структурировать его, выстроить в некоторой логической последовательности.

В последние годы в учебно-методической литературе по информатике стало традицией (проявление 4–7-го факторов!) организовывать учебный материал, начиная со сведений об информации и информационных процессах, т.е. с абстракций высокой степени общности, которые у детей младшего школьного возраста еще не могли сформироваться. В этом случае обучение оказывается ориентированным на заучивание материала без его понимания, так как не учитываются возрастные особенности формирования мышления детей.

Вообще существуют два альтернативных способа организации учебного материала: дедуктивный и индуктивный. Дедуктивный способ основан на том, что вначале вводятся абстрактные понятия, а затем они иллюстрируются, доказываются или обосновываются, рассматриваются их приложения и применения. Этот способ помогает сфокусировать внимание на абстракциях и предполагает преимущественно теоретический характер обучения, связан с доминирующим положением учителя как основного источника учебной информации. Дедуктивная организация материала позволяет за короткое время охватить сложные учебные темы. Второй способ — индуктивный — основан на построении материала “от практики”, от большого числа примеров и применений, к общению и постепенному “восхождению” до абстрактных понятий. Такой подход фокусирует внимание на процессах — мышления, логического вывода, учебной деятельности учащихся. Естественно, при индуктивном обучении легче может быть реализован личностно-ориентированный подход.

В младшей школе индуктивный подход к организации обучения в силу возрастных особенностей учащихся не имеет альтернатив, поскольку абстрактное мышление учащихся не сформировано, а его формирование — одна из задач обучения. Учет этого требования при обучении информатике в начальной школе диктует такую организацию учебного материала, которая начиналась бы со знакомства с предметами, конкретными объектами, затем действиями с ними, и в конечном итоге вела бы к абстрактным понятиям.

Так, начинать обучение целесообразно с модуля 2, “Компьютер и его устройства. Программные средства”. Этот модуль фиксирует внимание учащихся на непосредственном окружении — оборудовании кабинета вычислительной техники, компьютерах. В процессе развития предмета “информатика” в условиях недостатка компьютеров в школах, а там, где они были, разнообразия их моделей, сложился подход к разработке учебных курсов и пособий, не упоминающий о конкретных компьютерах, “задвигающий” сами компьютеры на задний план. Как уже говорилось, этот абстрактный подход неприменим в начальной школе. По нашему мнению, начинать обучение нужно именно со знакомства с компьютерами. Вполне естественно, что в условиях обучения в компьютерном классе первое, что привлечет внимание детей, будут именно компьютеры — и противостоять логике познания мира

ребенком, пытаться переключить его на абстрактные категории информации и информационного процесса было бы неоправданным насилием над психикой ребенка. Затем, когда компьютер утратит свою новизну (а это произойдет очень быстро!), необходимо переключиться на применение компьютеров — информационно-коммуникационные технологии, деятельность учащихся в информационной среде — это модуль 3. По времени освоение информационных технологий будет более длительным и займет значительную часть курса. Наконец, на базе освоения способов обработки информации с помощью компьютера происходит переход к обобщающим понятиям информации и информационных процессов, рациональным приемам обработки информации человеком с помощью мышления — модулю 1.

Заслуживает внимания и тот факт, что внимание и интерес учащихся начальной школы еще неустойчивы, склонны к самопроизвольному переключению, объем изучаемого материала ограничен, усвоение непрочное и требует периодического повторения. Полный курс информатики может быть построен в расчете на несколько лет обучения на основе концентров — годовых логически завершенных курсов, которые опирались бы на курсы предыдущих лет обучения. Изучение сложных тем целесообразно дробить на небольшие части, возвращаясь к ним на последующих этапах обучения — в следующих концентрах.

Предлагаемый курс рассчитан на 4 года обучения и имеет именно такую концентрическую структуру. На протяжении всех четырех лет основные изучаемые темы даются небольшими порциями, каждый год происходит возврат и повторение темы, продвижение вперед.

Представленный в комплексе (учебник + рабочая тетрадь + электронное приложение на CD) материал избыточен по отношению к любому учебному плану. Так, в рабочей тетради имеется расширенный (по сравнению с учебником) набор заданий для самостоятельной работы, а в электронном приложении на CD, помимо большого количества тренажеров, дублируется в конспективном виде часть учебной информации из учебника в форме текста и медиалекций. Это значит, что буквальное “прохождение” учебного комплекса не предусматривается. Избыточность материала позволяет учителю вести курс вариативно, самостоятельно выстроить занятия, учитывать особенности контингента учащихся и их индивидуальные запросы и возможности, комбинируя групповую работу над учебником и тетрадью, индивидуальную работу с компьютером в классе, самостоятельную работу учащихся дома. Комплекс предполагает высокую степень свободы и творчества учителя.

Итак, мы осветили, насколько это было возможно, характерные для любых методических рекомендаций вопросы “Зачем учить?” и “Чему учить?”. Далее мы обратимся к вопросу “КАК УЧИТЬ?” [9].

## 2. Как учить детей информатике в начальной школе?

Обучение может вестись в рамках самостоятельного предмета “Информатика, информационно-коммуникационные технологии” либо быть частью интегрированных

образовательных областей за счет региональной или школьной компонент учебного плана.

Занятия с детьми проводятся один раз в неделю по подгруппам в количестве 10–12 человек.

Рекомендуется проводить занятия в кабинете вычислительной техники, активно используя имеющееся компьютерное оборудование, стандартное программное обеспечение, а также CD-диск, входящий в поставочный комплекс.

Качественная реализация программы курса возможна при наличии полного программно-методического комплекса, а также соответствующей подготовки педагогов к его применению в педагогической деятельности.

Структура комплекса по учебной ступени 1–4-х классов содержит основные (взаимосвязанные) составляющие:

- Настоящие методические рекомендации, включающие программу курса и являющиеся системообразующим элементом программно-методического комплекса;
- учебное пособие (учебник) для моделирования познавательной деятельности учащихся;
- учебную тетрадь, включающую комплекс практических заданий;
- электронное приложение, содержащее инструментальный для моделирования самостоятельной деятельности учащегося на компьютере и средства мониторинга этой деятельности учителем.

Рассмотрим подробнее структуру учебного комплекса “Мир информатики”.

Комплекс по информатике для каждого года обучения в начальной школе “Мир информатики” содержит три упоминавшихся модуля, которые получили названия: “Здравствуй, класс компьютерный!” (посвященный компьютеру и программным средствам); “Поиграем, порисуем?” (посвященный информационно-коммуникационным технологиям); “Мир, в котором мы живем” (посвященный понятию информации и информационным процессам).

В учебнике даны тексты для чтения и вопросы к прочитанному, задания, понятия и правила. Вопросы размещены в овальных фигурах синего цвета, понятия и правила приведены в овальных фигурах зеленого цвета. Не следует требовать от учеников механического заучивания правил и определений. Они должны быть освоены на понятийном уровне.

Для закрепления и усвоения материала используется рабочая тетрадь. В ней предусмотрены задания, при выполнении которых потребуются ножницы, клей и цветные карандаши. Одна из задач данного вида упражнений заключается в развитии мелких мышц руки ученика.

К каждому занятию имеется мультимедийный материал, представленный на CD-диске “Мир информатики”, 6–9 лет. В помощь учителю для проведения уроков на диске представлено 16 медиалекций, 12 тренажеров, 10 различных развивающих игр, 150 иллюстраций.

В методическом пособии предложена определенная структура урока, которая может быть изменена в зависимости от индивидуальных способностей и возможностей учеников класса, педагогического мастерства учителя.

Наличие домашнего задания является необязательным условием, но, на наш взгляд, оно желательно, так как дети за недельный срок могут забыть изученный на уроке материал. Домашнее задание должно быть несложным, может носить творческий характер. Его задачей является подготовить детей к предстоящему уроку информатики. В качестве домашнего задания могут быть предложены упражнения из рабочей тетради или учебника.

Для проведения физкультминутки в пособии часто используется упражнение “Зарядка”. Однако учитель сам может выбрать физкультминутку, которая, на его взгляд, в большей степени сочетается с содержанием урока. В приложении 1 к пособию предложены возможные варианты физкультминуток, которые обогатят методическую копилку учителя.

При подготовке к занятиям учитель может дополнять и расширять материал, который предлагается в данном пособии, учитывая возникающие проблемы, интересы и потребности учащихся.

На последнем занятии каждой учебной четверти рекомендуется проводить урок-обобщение “Чему мы научились”. Во время обобщающего урока можно использовать сказочные сюжеты, игровые персонажи, игры-соревнования, что позволит в непринужденной форме вспомнить, повторить и закрепить полученные теоретические знания, практические умения и навыки работы с персональным компьютером.

## Литература

1. Мир информатики: Базовое учебное пособие для первого года обучения / Под ред. А.В. Могилева. Смоленск: Ассоциация XXI век, 2003, 80 с.
2. Мир информатики: Базовое учебное пособие для второго года обучения / Под ред. А.В. Могилева. Смоленск: Ассоциация XXI век, 2004, 88 с.
3. Мир информатики: Базовое учебное пособие для третьего года обучения / Под ред. А.В. Могилева. Смоленск: Ассоциация XXI век, 2005, 120 с.
4. Мир информатики: Рабочая тетрадь для первого года обучения / Под ред. А.В. Могилева. Смоленск: Ассоциация XXI век, 2003, 56 с.
5. Мир информатики: Рабочая тетрадь для второго года обучения / Под ред. А.В. Могилева. Смоленск: Ассоциация XXI век, 2004, 58 с.
6. Мир информатики: Рабочая тетрадь для третьего года обучения / Под ред. А.В. Могилева. Смоленск: Ассоциация XXI век, 2005, 64 с.
7. Мир информатики: CD 1–2-й год обучения / Под рук. А.В. Могилева. М.: Кирилл и Мефодий, 2002.
8. Мир информатики: CD 2–3-й год обучения / Под рук. А.В. Могилева. М.: Кирилл и Мефодий, 2003.
9. Могилев А.В., Булгакова Н.Н. Методические рекомендации к учебному комплексу “Мир информатики”. Смоленск: Ассоциация XXI век, 2005, 144 с.

### ПРИМЕРНОЕ ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ УРОКОВ ИНФОРМАТИКИ В ПЕРВЫЙ ГОД ОБУЧЕНИЯ

(из расчета 1 час в неделю, 34 учебных занятия)

№ п/п	Название темы	Кол-во часов	Страницы учебника	Номера тетради	Материалы CD-диска
<b>I четверть (9 часов) <i>Наш компьютер — верный друг</i></b>					
1	Здравствуй, класс компьютерный. Правила поведения в кабинете информатики	1	3–5		МЛ “Правила поведения в кабинете информатики”
2	Наш компьютер — верный друг	1	6–8	4	МЛ “Применение компьютеров”
3	Компьютер — электронная машина	1	9–10	1, 2, 3	МЛ “Применение компьютеров”
4	Основные устройства компьютера. Монитор	1	11–15	5, 6	МЛ “Компьютер и его основные устройства”
5	Системный блок	1	16	8–11	МЛ “Компьютер и его основные устройства”
6	Клавиатура	1	17–18	12–14, 20	МЛ “Клавиатура, работа на клавиатуре”
7	Мышь	1	19–20	7, 15, 19	МЛ “Мышь. Пиктограммы”; ТР “Мышь”
8	Указатели и стрелка. Щелчок, двойной щелчок	1	21–23	16, 18, 21	ТР “Мышь”
9	Чему мы научились. Обобщающий урок	1	23–24	17	ТР “Работа на клавиатуре”; ТР “Мышь”
<b>II четверть (7 часов) <i>Поиграем, порисуем</i></b>					
1	Чем мы можем рисовать. Палитра. Получение дополнительных цветов	1	25–28	22	МЛ “Графика”; РИ “Головоломка”
2	Раскрашивание. Контур	1	28–32	24, 29	МЛ “Раскрашивание компьютерных рисунков”; ОП “Раскрашивание компьютерных рисунков”
3	Пиктограммы. Конструкторы	1	32–37	24, 28	МЛ “Мышь. Пиктограммы”; ОП “Конструктор”

№ п/п	Название темы	Кол-во часов	Страницы учебника	Номера тетради	Материалы CD-диска
4	Конструирование	1	37–38	30	МЛ "Конструирование"; ОП "Конструктор"
5	Клавиатурные тренажеры	1	38–41	31	МЛ "Гимнастика для рук"; ТР "Работа на клавиатуре"
6	Как работать с клавиатурным тренажером?	1	40–42	25, 27	МЛ "Гимнастика для рук"; ТР "Работа на клавиатуре"
7	Чему мы научились	1	41–42	23, 25–27	Материалы раздела "Информационные технологии" на CD-диске
<b>III четверть (10 часов). Мир, в котором мы живем</b>					
1	Информация в нашей жизни	1	43–45	32–35	МЛ "Информация вокруг нас", МЛ "Понятие информации"; ТР "Работа на клавиатуре"
2	Как мы получаем информацию	1	46–48	36, 38, 41, 42	МЛ "Как мы получаем информацию", МЛ "Органы чувств и виды информации"; задания раздела "Информационные технологии"
3	Способы передачи информации	1	50–53	44, 46, 47	МЛ "Способы представления и передачи информации"; ОП "Игры с буквами, словами и цифрами"
4	В мире информации	1	45, 49, 50, 54–55	37, 39, 43, 45	МЛ "Органы чувств и виды информации"; ОП "Игры с буквами, словами и цифрами"
5	Суждение: истинное и ложное	1	56–59	48–52	МЛ "Элементы логики. Суждение: истинное и ложное"; ОП "Истинность и ложность суждений"
6	Сопоставление	1	60–63	54–56, 58	МЛ "Элементы логики. Сопоставление"; ОП "Сопоставление"
7	Причины и следствия	1	63–65	59–62	ОП "Сопоставление"
8	Как человек усваивает информацию	1	45, 58, 62	40, 53, 57	ТР "Работа на клавиатуре"
9	Множества	1	67–69	63–65	МЛ "Множества", ОП "Множества"
10	Чему мы научились	1	С. 45, 66	66	
<b>IV четверть (8 часов). Исполнители и системы команд</b>					
1	Правила в жизни человека	1	70–71	67– 70, 72	МЛ "План и правила" (слайды 1–6), ТР "Работа на клавиатуре"
2	Правила в жизни человека (продолжение)	1	71–73	71, 73	ТР "Работа на клавиатуре"
3	План	1	73–75	76–77	ОП "Головоломка" раздела "Информационные технологии" темы "Графика"
4	Головоломки	1	75	74–75, 78	ОП "Конструктор" раздела "Информационные технологии" темы "Конструирование"
5	Исполнитель	1	76–77	79–82	МЛ "Исполнитель", РИ "Исполнитель Транспортер"
6	Исполнитель (продолжение)	1	78	83–86	МЛ "Пример исполнителя", РИ "Исполнитель Транспортер"
7	Обобщение. Чему мы научились	1			ОП "Головоломка" раздела "Информационные технологии" темы "Графика", РИ "Исполнитель Транспортер"
<p>МЛ — медиалекция;    РИ — развивающая игра;  ТР — тренажер;    ОП — обучающая программа.</p>					

# В мир информатики

# 67 (16—31 января)

Газета для пытливых учеников  
и их талантливых учителей

## Как решать задачи?

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

*От редакции.* Одним из важных условий, выполнение которых позволит вам, уважаемый читатель, стать хорошим программистом, является большой опыт решения задач по программированию. Решайте задачи, побольше и разные — абстрактные и содержательные, “на 5 минут” и “на день работы”. Все это обязательно пригодится. Поверьте, что много решенных задач — не бывает. При решении вам и понадобятся советы, сделанные в статье Н.М. Тимофеевой.

Стремление решить задачу заложено  
в самой природе человека.

Д.Поля

Для чего нужно разрабатывать программы? Прежде всего для того, чтобы поручить компьютеру решить задачу, причем часто такую, которую человек может решить и сам. Только компьютер сделает это быстрее. Конечно, надо “объяснить” компьютеру, как именно надо решать задачу. А для того чтобы объяснить, как решать (хоть компьютеру, хоть соседу по парте), надо самому знать, как. Очень понятно о решении задач писал известный математик Джордж Поля. Родился он в Венгрии в 1887 году, преподавал в Швейцарии и в США. Его имя и фамилию произносят по-разному в разных странах. Так, в Венгрии он известен как Дьердь Поля, в Германии — как Георг Поля, а в США — как Джордж Поля.

Поля писал, что решение задач — это *практическое* искусство, подобно плаванию, или катанию на лыжах, или игре на пианино: вы можете научиться этому, только практикуясь. Он сравнивал, что если вы захотите научиться плавать, то вынуждены будете зайти в воду, а если вы захотите стать человеком, хорошо решающим задачи, вы вынуждены их решать.

Поля был сторонником применения технологий решения задач при изучении математики. Более всего он известен благодаря своим выдающимся книгам “Как решать задачу” и “Математика и правдоподобные рассуждения” с примерами, доступными школьникам.

В кратком изложении идеи его книг можно выразить следующим образом.

### 1. Понимание задачи

Прежде всего вы должны *понять* задачу.

- Что неизвестно? Что дано? Что является условием?
- Возможно ли выполнить условие? Условие достаточно для определения неизвестных величин? Или оно недостаточно? Или избыточно? Или противоречиво?
- Сделайте рисунок. Введите соответствующие обозначения.
- Разделите условие задачи на части. Вы можете записать их?

### 2. Разработка плана

- Определите связь между исходными данными и неизвестным. Если прямая связь не может быть найдена, возможно, вам придется рассмотреть вспомогательную задачу. В итоге вы должны получить *план решения*.
- Встречалась ли вам такая задача раньше? Может быть, вы сталкивались с подобной задачей в несколько другом виде?
- Вы знаете похожую задачу? Известна вам теорема или формула, которая может быть применена?
- Посмотрите на искомую величину! И постарайтесь вспомнить знакомую задачу с таким же или похожим неизвестным.
- Вот задача, похожая на вашу и решенная раньше. Можете вы воспользоваться ею? Можете применить ее результат? Можете воспользоваться способом решения этой задачи? Можете ввести некоторые вспомогательные элементы, чтобы можно было использовать решенную задачу?
- Могли бы вы переформулировать задачу? Могли бы вы сформулировать ее еще по-другому? Вернитесь к условию задачи.
- Если вы не можете решить предлагаемую задачу, постарайтесь сначала решить какую-нибудь подобную задачу. Не могли бы вы представить более понятную подобную задачу? Более общую задачу?

Школа программирования

Более частную задачу? Аналогичную задачу? Можете решить часть задачи? Оставьте только часть условия, отбросьте другую часть; насколько неизвестная величина тогда будет определена, как можно ее изменить? Можно ли извлечь что-либо полезное из исходных данных? Не могли бы вы представить себе другие данные, подходящие для нахождения неизвестного. Не могли бы вы изменить неизвестное и исходные данные или и то и другое, если необходимо, для того чтобы новое неизвестное и новые исходные данные стали ближе друг к другу?

- Все исходные данные вы применили? Вы полностью применили условие? Все математические понятия, включенные в задачу, вы приняли во внимание?

### 3. Выполнение плана

- Выполните свой план.
- Проверяйте каждый шаг, выполняя свой план решения. Вам ясно, что шаг верен? Можете доказать, что он безошибочен?

### 4. Взгляд назад

- Проверьте полученное решение.
- Можете проверить результат? Можете проверить ход решения?
- Можете решить по-другому? Можете понять это сразу?
- Можете применить этот результат или способ решения для какой-нибудь другой задачи?

Другими словами, идею книги Джорджа Пойа “Как решать задачу” можно выразить так: понять, запланировать, выполнить, проверить.

Далее мы расскажем, как о технологии решения задач по Пойа применительно к программированию пишет Грэм Саммерс в своем учебнике “Программирование на Visual Basic”. Эта технология в учебнике используется в каждом примере программирования. В ближайших выпусках газеты “В мир информатики” будут рассмотрены несколько примеров из книги, поэтому знакомство с указанной технологией будет для вас полезным. Итак, отрывки из книги Г.Саммерса.

Джордж Пойа преподавал математику в Станфордском университете (Калифорния). Он известен своими статьями о решении задач. Однажды он написал: “Пытаясь решать задачи, вы должны наблюдать и повторять то, что делают другие, когда решают задачи, и в конце концов вы учитесь решать задачи, решая их”.

В течение этого курса<sup>1</sup> вы будете наблюдать и повторять, но, что более важно, вы будете решать

задачи самостоятельно. В своей книге под названием “Как решать задачу” Пойа описал шаги, необходимые для решения задачи:

- 1) определить задачу;
- 2) запланировать решение;
- 3) выполнить решение;
- 4) оглянуться назад.

### 1. Определение задачи

Когда требуется решить задачу, первое, что должно быть сделано, — исследовать ее. Вы должны установить, что дано, что нужно сделать. Надо быть уверенным, что задача решается, иначе последующие шаги могут состоять из напрасных усилий.

### 2. Планирование решения

После исследования задачи рассматриваются шаги, которые требуются для решения<sup>2</sup>, и порядок, в котором они должны быть выполнены. Шаги, которые необходимы для решения задачи и их последовательность, — это и есть *алгоритм*<sup>3</sup>. Такое определение может использоваться и как компьютерный термин, и в обычной жизни. Например, существуют алгоритмы для вязания джемперов, выполнения умножения столбиком, выпекания хлеба и т.д. В компьютерной терминологии планирование решения означает *разработку алгоритма*.

### 3. Выполнение решения

В книге выполнение решения означает описание алгоритма на языке, понятном компьютеру. Такое описание называется *программой*, а язык, которым она выражена, — *языком программирования*. Как только программа написана, мы заставим компьютер ее выполнить. На стадии выполнения будет разумно точно документировать принятые шаги, чтобы легче было проверять или исправлять их в последующем.

### 4. Взгляд назад

Мы сначала должны убедиться, что процесс на последнем шаге выполнения соответствует заданию, которое было определено первоначально. Если это не так, тогда надо просмотреть все предыдущие шаги. Когда мы будем удовлетворены тем, как выполняются заданием, мы можем снова вернуться к предыдущим шагам для их улучшения. И так этот цикл может повторяться несколько раз, пока мы не будем удовлетворены полностью. В компьютерной терминологии такой цикл называют *циклом разработки программного обеспечения*.

Рассмотрим несколько примеров алгоритмов:

<sup>2</sup> Для этого, конечно, надо ответить на вопросы, на которые обращает внимание Д.Пойа (см. выше по тексту). — **Ред.**

<sup>3</sup> Возможно, что на уроках информатики вы рассматривали другое определение понятия “алгоритм”. — **Ред.**

<sup>1</sup> Речь идет о курсе, обсуждаемом на страницах книги Г.Саммерса. — **Ред.**

Процесс (задача)	Алгоритм	Процессор <sup>4</sup>
Вязание джемпера	Образец вязания	Руки человека или вязальная машина
Выпекание хлеба	Рецепт	Пекарь
Сборка модели автомобиля	Инструкция по сборке	Моделист-конструктор

Во всех этих случаях процессор — это тот, кто выполняет простые операции простыми инструментами. Процессор достигнет желаемого результата при условии, что алгоритм точно удовлетворяет условиям задачи и выполняется без ошибок.

Многие процессы взаимодействуют с окружающей обстановкой, осуществляя *ввод* и обеспечивая *вывод*. В приведенных выше примерах шерсть, мука и пластиковые детали автомобиля представляют собой ввод, тогда как выводом являются джемпера, хлеб и модели. Подробное описание ввода и вывода — составная часть описания алгоритма. Если ввод будет неправильным, то процесс может быть невыполнимым или вывод будет ошибочным или нежелательным. Например, если при выпекании хлеба ввод состоял в стиральном порошке вместо муки, то и вывод будет чем-нибудь менее съедобным, чем булка хлеба.

Все процессы в конце концов завершаются: получается джемпер, хлеб и т.п. Однако, кроме этого, есть и задачи, которые не являются достижимыми в данный момент, например, осуществление мира во всем мире.

Все компьютерные процессы, которые мы рассмотрим в этой книге, имеют конечный результат. То есть алгоритмы имеют конечную точку, достижение которой предполагает, что требуемое задание выполнено.

### Ошибки

Существуют три основных типа ошибок, которые могут проявиться при попытке выполнить процесс. Первый тип — *синтаксические ошибки*. Это ошибки в записи алгоритма, которые делают его непонятным для исполнителя. Это может быть ошибка правописания или символ, который не может быть переведен, например, инструкция “добавить 20@0 грамм простой муки”.

Второй тип ошибок — *ошибки исполнения*. Эти ошибки проявляются тогда, когда исполнитель не может выполнить инструкцию. Например, ошибка исполнения проявится, если по инструкции надо прибавить 2 литра молока в литровую банку. Другая ошибка может заключаться в попытке поделить число на ноль. Хотя инструкции правильны по синтаксису, но их невозможно выполнить. К ошибкам исполнения часто приводят неверные данные.

<sup>4</sup> Ясно, что термин “процессор” здесь не имеет непосредственного отношения к процессору компьютера, а связан с понятием “исполнитель алгоритма”. — Ред.

Третий тип ошибок — *логические ошибки*. Этот тип ошибок труднее всего определить, поскольку при отсутствии синтаксических ошибок и ошибок исполнения программа выдает неожиданные или неверные результаты. Для проверки правильности логики (отсутствия логических ошибок) программу нужно протестировать с использованием *проверочных данных*.

### Вопросы для проверки изученного

1. Какие 4 главных шага надо сделать при решении задачи?

2. Дайте определение понятия “алгоритм”.

3. Почему при решении задачи так важен последний шаг — “взгляд назад”?

4. Ниже представлены примеры процессов.

I. Выпекание хлеба.

II. Установка вашего видеомэгагнитофона на запись в ваше отсутствие.

III. Выполнение умножения двух чисел столбиком.

Для каждого случая ответьте на следующие вопросы:

а) в какой форме обычно выражается соответствующий алгоритм;

б) приведите примеры простых шагов, на которые может быть разделен алгоритм;

в) что является вводом;

г) что является выводом;

д) в чем будет заключаться “взгляд назад”?

5. Приведите несколько примеров неправильного ввода для следующих процессов:

— изготовление модели планера из древесины;

— функционирование уличного телефона;

— деление одного целого числа на другое.

6. Какие из следующих процессов могут завершиться:

— вымешивание теста для хлеба;

— деление числа 3 на 11 столбиком;

— поиск лох-несского чудовища;

— приготовление обжаренных в масле овощей?

При каких обстоятельствах это может произойти?

7. Определите разницу между:

— синтаксической ошибкой и ошибкой исполнения;

— ошибкой исполнения и логической ошибкой.

### Литература

1. *Summers Graeme*. Programming with Visual Basic. Second Edition (Visual Basic version 6). Nelson, 1999.

2. *Поля Д.* Как решать задачи: Пособие для учителей. Перевод с англ. М.: Учпедгиз, 1959.

3. <http://www.mathgym.com.au/htdocs/polyab.htm>.

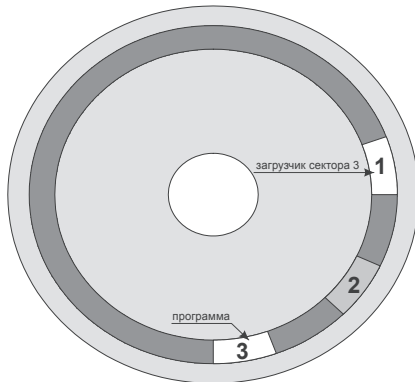
## Работает без операционной системы (!)

Е.А. ЕРЕМИН,  
г. Пермь

Окончание. Начало см. газету-вкладку  
“В мир информатики” № 65 (“Информатика” № 24/2005)

Напомним, что в первой части статьи были рассмотрены теоретические вопросы, которые позволяют провести эксперимент по созданию и запуску программы, которая будет работать без загрузки операционной системы.

Составим план нашего эксперимента. Прежде всего мы имеем код программы [1], которую намереваемся запустить. Учитывая небольшой размер, ее можно было бы поместить непосредственно в загрузочный сектор. Тем не менее для большего реализма предлагаю поступить немного сложнее: программу записать в произвольный сектор, например, с номером 3, а в секторе 1 разместить загрузчик этого сектора (см. рисунок).



**Внимание!** Для простоты формирования кода мы не будем заботиться о совместимости полученной нами дискеты с MS-DOS или Windows. Поэтому для экспериментов потребуется дискета, не содержащая никакой информации! После завершения экспериментов дискету придется заново отформатировать.

Для занесения подготовленного кода загрузчика и программы исполнителя воспользуемся программой на языке Паскаль; ее полный текст приводится в приложении.

Программа содержит два массива данных строкового типа — *code0* и *code2*. В первом хранится программный код загрузчика, во втором — код программы. Обращаем внимание на тот факт, что в массивах содержатся строки двух видов — коды инструкций программы и тексты сообщений, которые начинаются с кавычек. В связи с

этим при их занесении в буфер<sup>5</sup> для записи на диск предусмотрены различные ветви оператора *If*: символы сообщений просто по одному копируются в буфер, а шестнадцатеричные (признаком этого служит символ “\$” перед ними) и десятичные числа сначала из строкового представления процедурой *Val* преобразуются в целое число, которое затем с помощью функции *Chr* становится символом. В итоге формально вся наша кодовая программа образует однородный массив символов, который и заносится в требуемый сектор дискеты.

**Примечание.** Вполне возможно оптимизировать программу, объединив массивы *code0* и *code2* в один двумерный массив. При этом основная программа сократится почти вдвое за счет применения оператора цикла. Предлагаем читателю проделать данную оптимизацию самостоятельно.

Запись на диск (или чтение с него) осуществляет ассемблерная процедура *Disket*. У нее два параметра: *r* определяет режим работы (*Rread* — чтение, *Rwrite* — запись), а *s* задает номер сектора (в нашем случае 1 или 3). Процедура заполняет все необходимые регистры микропроцессора в соответствии с таблицей, приведенной в конце первой части статьи, а затем вызывает процедуру ROM BIOS, используя инструкцию *INT 13*. После завершения обмена для контроля выводится информация о наличии или отсутствии ошибок (последний случай соответствует нулевому коду).

Рекомендуем читателям перед набором текста программы внимательно разобрать ее работу. Автор считает, что знаний в объеме школьного курса программирования должно быть для этого вполне достаточно.

Завершая обсуждение программы, прокомментируем ее часть в машинных кодах, т.е. то, что с точки зрения данной Паскаль-программы является просто информацией для записи в сектора 1 и 3.

Загрузчик (*loader*), формируемый с помощью массива *code0*, производит следующие несложные действия:

- запрашивает текущее положение курсора на экране;
- начиная с этого положения, выводит текст “LOADING...” (разумеется, данные действия не несут особой функциональности);
- читает сектор 3 дискеты и записывает данные в оперативную память, начиная с произвольно выбранного адреса 7E00h (BIOS всегда грузит boot-сектор с адреса 7C00h);
- запускает то, что прочитано и записано<sup>6</sup>.

<sup>5</sup> Роль которого выполняет массив *buf*. — Ред.

<sup>6</sup> Запуск программы производится последними тремя командами; к сожалению, их разъяснение требует некоторых дополнительных сведений, которые для краткости опущены.



Загруженная программа совершенно аналогично выводит на экран текст "READY" и начинает работу.

Теперь можно нажимать клавиши **[+]** или **[2]** — в ответ программа будет либо увеличивать число на экране на 1, либо удваивать его. Как только результат превысит 99, предпоследняя команда условного перехода не сработает, а процессор, выполняя инструкцию остановки `hlt`, прекратит работу по нашей программе и "замрет" в ожидании перезагрузки.

Осталось раскрыть последний секрет. У наиболее внимательных читателей наверняка возник вопрос: а откуда известно, что коды программы должны быть именно такими? Как узнать, каков, например, код операции `mov ah, 3` или любой другой, входящей в нашу программу? Для этой цели автор пользовался стандартным отладчиком `Debug`, который способен воспринимать набор на ассемблере, преобразовывать его в машинный код и при желании вывести полученный код на экран. Этот процесс может служить темой для самостоятельного подробного разговора, поэтому в данной статье ограничимся тем, что уже сказано. Если же вам хотелось бы познакомиться с работой программы `Debug`, напишите об этом в редакцию.

Автор надеется, что все, кто решился на проведение нашего "дерзкого" эксперимента, сумели разобраться в устройстве программы. Как действовать дальше?

1. Очень внимательно наберите программу и тщательно проверьте. Сохраните текст Паскаль-программы на диске, прежде чем что-либо делать дальше.

2. Вставьте в дисковод чистую дискету или дискету, не содержащую информации.

3. Запустите программу на исполнение. Проследите, чтобы во время исполнения последовало обращение к дисководу.

4. Найдите в тексте программы на с. 43 "закомментированную" строку, читающую третий сектор (ее легко опознать по тексту `disket(Rread, 3)`). Удалите фигурные скобки — признак комментария — и запустите программу еще раз. Вы увидите на экране коды, записанные в секторе 3. Если вы все сделали правильно и программа действительно туда записалась, вы увидите на экране следующую последовательность чисел:

180 3 183 0 205 16 184 1 19 179 2 185  
15 0 141 46 73 126 205 16 185 10 и т.д.

(нетрудно убедиться, что это коды первых команд, представленные в десятичном виде).

5. Итак, дискета сформирована и готова к проведению эксперимента. Не вынимая ее из дисковода, обычным образом перезагрузите компьютер. При условии, что флоппи-дисковод установлен в качестве первого устройства загрузки, а в набранном коде нет ошибок, вы увидите свои загрузочные сообщения и сможете поработать с программой-вычислителем (без всякой Windows!).

Автор надеется, что интеллектуальное удовлетворение, которое вы получите от запуска программы в столь экзотических условиях, скомпенсирует все приложенные усилия. Возможно, некоторым из вас захочется сделать свою собственную программу, способную работать на компьютере без участия Билла Гейтса и его команды (©). Попробуйте обязательно и, если получится что-то интересное, напишите в редакцию.

### Литература

1. Еремин Е.А. Система команд ЭВМ. / Информатика № 9/2005, с. 23—30 (см. п. 4).

### Приложение

#### Текст программы формирования загрузочного диска

```

Program Make_Disk;
Uses CRT;
Type regime = (Rread, Rwrite);
Const crlf = Chr(13) + Chr(10);
n0 = 41; {Количество байт}
code0: array [1 .. n0] of string = {loader}
('b4', '3', {mov ah, 3}
'b7', '0', {mov bh, 0}
'cd', '$10', {int 10h — положение курсора в dx}
'b8', '1', '$13', {mov ax, 1301h}
'b3', '2', {mov bl, 2}
'b9', '$e', '0', {mov cx, 0eh}
'8d', '$2e', '$28', '$7c', {lea bp, [7c28h] — адрес сообщения}
'cd', '$10', {int 10h — вывести "LOADING"}
'b8', '1', '2', {mov ax, 0201h}
'ba', '0', '0', {mov dx, 0}
'b9', '3', '0', {mov cx, 3}

```

```

    '$8d', '$1e', '0', '$7e',    {lea bx, [7e00h] - адрес программы}
    '$cd', '$13',                {int 13h - читать сектор 3}
    '$b8', '0', '$7e',          {mov ax, 7e00h}
    '$50',                       {push ax}
    '$c3',                       {ret - выполним прочитанное}
    ''' + crlf + crlf + 'LOADING...');
n2 = 74; {Количество байт}
code2 : array [1 .. n2] of string = {program}
    ('$b4', '3',                 {mov ah, 3}
    '$b7', '0',                 {mov bh, 0}
    '$cd', '$10',               {int 10h - положение курсора в dx}
    '$b8', '1', '$13',          {mov ax, 1301h}
    '$b3', '2',                 {mov bl, 2}
    '$b9', '$f', '0',          {mov cx, 0fh}
    '$8d', '$2e', '$49', '$7e', {lea bp, [7e49h] - адрес сообщения}
    '$cd', '$10',               {int 10h - вывести "READY"}
    '$B9', '$A', '0',          {mov cx, 10 - константа}
    '$31', '$DB',              {xor bx, bx - X = 0}
    '$30', '$C0',              {xor al, al - старшая цифра = 0}
    '$30', '$D2',              {xor dl, dl - младшая цифра = 0}
    '$0C', '$30',              {@1: or al, 30h}
    '$B4', '$0E',              {mov ah, 0eh}
    '$CD', '$10',              {int 10h - вывести старшую цифру}
    '$88', '$D0',              {mov al, dl}
    '$0C', '$30',              {or al, 30h}
    '$CD', '$10',              {int 10h - вывести младшую цифру}
    '$B0', '$20',              {mov al, ' '}
    '$CD', '$10',              {int 10h - вывести пробел}
    '$B4', '0',                 {@2: mov ah, 0}
    '$CD', '$16',              {int 16h - ввести символ с клавиатуры}
    '$3C', '$32',              {cmp al, '2'}
    '$74', '7',                 {jz @3 - если введена '2'}
    '$3C', '$2B',              {cmp al, '+'}
    '$75', '$F4',              {jnz @2 - если не '+', то повторить ввод}
    '$43',                       {inc bx - X + 1}
    '$EB', '2',                 {jmp @4}
    '1', '$DB',                {@3: add bx, bx - 2X}
    '$31', '$D2',              {@4: xor dx, dx}
    '$89', '$D8',              {mov ax, bx}
    '$F7', '$F1',              {div cx - деление на 10}
    '$39', '$C8',              {cmp ax, cx}
    '$7C', '$D5',              {jl @1 - повторить, если результат < 10}
    '$f4',                       {halt}
    ''' + crlf + 'READY! ' + crlf + crlf);
Var buf: array [1 .. 512] of char; i, ic, ib, k, q: integer;

Function Disket(r: regime; s: byte): boolean;
Var p, e, n: byte;
Begin
If r = Rwrite Then p := 3 Else p := 2;
ASM mov ah, 0
    mov dl, 0
    int 13h {Инициализация дисководов A:, полезна после ошибок}
    mov dl, 0 {Номер диска}
    mov dh, 0 {Номер стороны}
    mov ch, 0 {Номер дорожки}
    mov cl, s {Номер сектора}
    mov al, 1 {Число секторов}
    mov ah, p {Запись - 3, чтение - 2}
    lea bx, buf {Адрес буфера}
    int 13h {Запись или чтение секторов}

```

```

        mov n, al {Количество секторов}
        mov e, ah {Ошибки}
End; {ASM}
If r = Rwrite Then write('записано: ')
        Else write('прочитано: ');
writeln(n, ' сект., с N ', s); writeln('код ошибки: ', e);
Disket := (e = 0);
End; {Функции Disket}

Procedure ToBuf(c: char);
Begin
buf[ib] := c; ib := ib + 1
End;

BEGIN
clrscr; TextColor(15);
{Disket(Rread, 3);
For i := 1 To 80 Do write(Ord(buf[i]), ' '); writeln;}
For i := 1 To 512 Do buf[i] := Chr(0);
ic := 1; ib := 1;
While ic <= n0 Do {loader}
Begin
If code0[ic][1] = '' Then {Текст сообщения}
        For q := 2 To Length(code0[ic]) Do toBuf(code0[ic][q])
Else {Код инструкции программы}
Begin
Val(code0[ic], k, q);
ToBuf(Chr(k))
End;
ic := ic + 1;
End;
Disket(Rwrite, 1); {Запишем загрузчик в начальный сектор}
For i := 1 To 512 Do buf[i] := Chr(0);
ic := 1; ib := 1;
While ic <= n2 Do {program}
Begin
If code2[ic][1] = '' Then {Текст сообщения}
        For q := 2 To Length(code2[ic]) Do ToBuf(code2[ic][q])
Else {Код инструкции программы}
Begin
Val(code2[ic], k, q);
ToBuf(Chr(k))
End;
ic := ic + 1
End;
Disket(Rwrite, 3); {Запишем программу в сектор 3}
END.

```

## Задчник

### Однофамильцы в поезде

В поезде едут пассажиры Иванов, Петров и Сидоров. Оказалось, что такие же фамилии у машиниста поезда, его помощника и у проводника. Известно, что:

- 1) пассажир Иванов живет в Москве;
- 2) проводник живет на полпути между Москвой и Санкт-Петербургом;
- 3) пассажир — однофамилец проводника, живет в Санкт-Петербурге;

4) у пассажира, который живет ближе к месту жительства проводника, чем другие пассажиры, вдвое больше детей, чем у проводника;

5) у пассажира Петрова трое детей;

6) Сидоров (из поезда бригады) недавно выиграл у машиниста партию на бильярде.

Какая фамилия у машиниста поезда?

Ответ присылайте в редакцию.

**Буквенный ребус****“Ломаем” голову**

Определите, какие цифры зашифрованы буквами А, Г, Е, З, И, Н, О, П, Т, У, если

$$\begin{array}{r}
 \text{АТУ} + \text{ИАЗ} = \text{ИИТЕ} \\
 \text{---} \quad \quad \quad \text{---} \quad \quad \quad \text{---} \\
 \text{НЕГ} : \text{ИОГ} = \text{Е} \\
 \text{---} \quad \quad \quad \text{---} \quad \quad \quad \text{---} \\
 \text{ПАУ} - \text{НЗ} = \text{ППА}
 \end{array}$$

Одинаковыми буквами зашифрованы одинаковые цифры, разными буквами — разные цифры.

**Внимание!  
Конкурс****Конкурс № 42 для учащихся**

Ответьте, пожалуйста, на вопросы, приведенные в конце статьи Н.М. Тимофеевой “Как решать задачи?”, опубликованной в этом выпуске газеты.

Ответы отправьте в редакцию до 15 февраля по адресу: 121165, Москва, ул. Киевская, д. 24, “Первое сентября”, “Информатика” или по электронной почте [inf@1september.ru](mailto:inf@1september.ru).

Пожалуйста, четко укажите в ответе свои фамилию и имя, населенный пункт, номер и адрес школы, фамилию, имя и отчество учителя информатики.

**Итоги конкурса № 40 для учащихся****Продолжение. Начало см. в предыдущем номере нашей газеты**

Мы продолжаем публиковать примеры пословиц, поговорок, крылатых фраз и т.п., в которых имеется слово *один*.

Но сначала приведем две высказывания, которые прислала Айгуль Загафуранова, средняя школа деревни Сейтяк Балтачевского р-на, Республика Башкортостан (учитель Загафуранова А.Ф.):

Вычисление с помощью двоек... является для науки основным и порождает новые открытия... При сведении чисел к простейшим началам, как 0 и 1, везде появляется чудесный порядок.

Готфрид Вильгельм Лейбниц

В своей бинарной арифметике Лейбниц видел прообраз творения. Ему представлялось, что единица представляет божественное начало, а ноль — небытие и что высшее существо создает все существующее из небытия точно таким образом, как единица и ноль в его системе выражают все числа.

Пьер Симон Лаплас

Змея *один* раз в году меняет шкуру, а предатель каждый день.

*Один* сидит, сам-друг спит.

*Один* со страху помер, другой ожил.

*Один* собирает, другой зевает.

*Один* соврет, хоть кулаки суй; другой соврет, иголки не подбить.

*Один* солдат — не полк.

*Один* старый друг лучше новых двух.

*Один* сын — не сын, два — не кормилицы.

*Один* только бог видит (или слышит) нас. Только богу и плакаться.

*Один* ты в сапогах ходишь.

*Один* улей — улей, а пять — пасака.

*Один* ум — пол-ума; три ума — полтора ума; два ума — ум.

*Один* умный десять безумных водит. (Прислала Косячкова Марина.)

*Один* хлеб попу, одна радость — что свадьба, что похороны.

*Один* хлеба не съешь.

*Один* цветок весны не делает.

*Один* язык за три ума. (Привела Радзеновская Анастасия.)

*Один*, да дорог — не надобно и сорок.

*Один*, другой — обчелся. (Прислала Хаустова Кристина.)

*Один*, как бог, как перст, как порох в глазу, как верста в поле, как маков цвет.

*Один*, как перст, как маков цвет, как красное солнышко, как ясный месяц, как верста в поле.

Одна копейка рубль бережет, а *один* рубль голову стережет. (Привела Тюрикова Елена.)

Одно к одному. Сваливай на *один* загорбок!

Одно чадо, и то чало. *Один* сынище, и тот Фомище.

Одному началу *один* и конец.

Они одной школы, одной выучки, одной шерсти, масти, *один* черт.

От великого до смешного *один* шаг.

От добра до худа *один* шаток.  
 От любви до ненависти *один* шаг.  
 От худа до худа *один* шаток.  
 Первый парень на деревне, а в деревне *один* дом.  
 Плоха мышь, что *один* только лаз (одну лазею) знает.  
 Подчас и *один* стоит семерых.  
 Поп любит блин, а ел бы он *один*.  
 Поп любит блин, да чтоб не *один*.  
 Рубить семерым, а топор *один*.  
 Сапер (минер) ошибается только *один* раз.  
 Свет не то, что клин (земли) — *один* весь не захватит.  
 Семеро капралов, да *один* рядовой.  
 Семеро лежат в куче, а *один* всех растаскает.  
 Семеро не *один*, в обиду не дадим.  
 Семеро с пером, а *один* с топором.  
 Семь бед — *один* ответ.  
 Семь листов бумажных, *один* золотой.  
 Семь раз проверь, а один раз поверь. (Прислала Сенокосова Любовь.)  
 Семь сёл, *один* вол, да и тот гол.  
 Семьдесят приказчиков, *один* рядовой, да и то не свой (или чужой).  
 Семья воюет, а *один* — горюет.  
 Скопили (или: свели) домик в *один* уголок.  
 Скрипка да гудок сведет домик в *один* уголок.  
 Так ли, сяк ли — уж *один* конец.  
 То не страх, что вместе; а сунься-ка *один*.  
 Только и есть за душой. *Один* одним за душой.  
 Тот и господин, кто все может сделать *один*.  
 Трус умирает сто (тысячу) раз, а герой — *один* раз.  
 Ты у меня *один* одним, как синь порох в глазу.  
 У невесты женихов сто *один*, а достанется *один*.  
 У него *один* замок гостит, да и тот на пробое висит.  
 У пьяного семь клетей, а проспится — *один* плетень.  
 — Фекул, что губы надул? — Да кафтан прожег. — Зачинить можно? — Да иглы нет. — А велика дыра? — Да *один* ворот остался.  
 Фик-фок на *один* бок.  
 Хотя горшок и *один*, да сам себе господин.  
 Что не сможешь сделать *один*, сделают десятеро. (Прислал Аглямов Айнур.)  
 Через восемьдесят могил хватил *один* блин.  
 Через семьдесят могил хватил (разорвали) *один* блин.  
 Черная собака, белая собака — а все *один* пес.  
 Что два, то не *один*. То два, а то полтора.  
 Шел *один*, нашел пять рублей; трое пойдут, много ли найдут?  
 Это *один* только задаток, а дело впереди.

Основная часть перечисленных примеров была приведена несколькими участниками конкурса. В ответах были и оригинальные примеры. Мы публикуем их<sup>7</sup>.

Быстрота знает только *один* марш — вперед (А.В. Суворов).

Где *один* гриб, там и другой.

Кому трудно учиться *один* день, тому трудно будет всю жизнь.

К славе ведет *один* путь — труд.

Курит *один*, а дым его травит многих.

Лучше *один* день соколом, чем век — вороном.

*Один* мигает, а другой смеркает.

Полководец обсуждает со многими, а решает *один*.

Прислал Артур Абдрахманов

*Один* родился и тот не пригодился.

*Один* тянет за гриву, другой — за хвост.

Друга испытывают *один* раз.

Дважды подумай, *один* раз делай.

Привел Рамиль Абдуллин

Баба да бес — *один* у (в) них вес.

Благому чудотворцу Николаю два праздника в году, а Касьяну немилостивому *один* в четыре года.

Бог отстанет, так ни *один* не встанет.

Бредет и зауряд в *один* ряд.

В день Казанской весь червец собирается под *один* куст.

В семидесяти двух — *один* козий (песий) дух.

Вавила — красное рыло. Иван — болван. Андрей — ротозей. Фекул — губы надул. Пахом — вся рожка в *один* ком.

Все говорят, что по суду *один* будет виноват.

Все люди, как люди, *один* черт в колпаке.

Всяк за своих стоит, а *один* бог за всех.

Два брата родные, и оба Ивановичи, да *один* Дон, а другой Шат (т.е. *один* дельный, другой шатун. Реки Дон и Шат вытекают из Иван-озера).

Два Егорья: *один* холодный, другой голодный (26 ноября, 23 апреля).

Два мужа наружу да *один* в сундуке (от комедии).

Два Николы: *один* с травой, другой с морозом.

Два Николы; *один* травный, другой морозный.

Два таких круга или *один* тусклый — к морозу; красный круг — к ветру, перерванный — к снегу.

День св. Феодосии стоит *один* всех понедельников.

Калмыцкую лошадь *один* только калмык и перепрямит.

Когда *один* хлеб вынут раньше прочих и разрежут, то все хлебы испортятся.

<sup>7</sup> Если вы не найдете в перечне примеры, предложенные вами, пожалуйста, сообщите об этом в редакцию — мы их обязательно опубликуем с указанием вашей фамилии.

Кто смел, тот и бел. Кто смел, тот *один* все съел.  
Мир никем не судится, одним богом. Мир судит *один* бог.

На мир и суда нет. Мир *один* бог судит.

На Руси два Егорья: *один* холодный, другой голодный (26 ноября и 23 апреля).

На старости две радости: *один* сын — вор, другой — пьяница.

Назар все *один* слизал.

Нет ли вошки, нет ли блошки — *один* червячок, да и тот золотой: бить или на волю пускать? (Игра нянек.)

Новгород (древний) судит *один* бог.

*Один* глаз поперек (т.е. пьян).

*Один* Иван — должно; два Иван — можно; три Иван — никак не возможно (сказал немец про Ивана Ивановича Иванова).

*Один* крест хлеба не ест.

*Один* одного лучше (*один* одного хуже). Слово на подбор.

*Один* пошел — полтину нашел; семеро пойдут — много ли найдут?

*Один* проторил тропу, а все ходят.

*Один* раз (однова) перевозиться (домом) — полпожара выстоять; два раза (двою) перевозиться — весь пожар выстоять.

*Один* рычит да лает, другая мурлычет да фыркает.

*Один* скачет, *один* и плачет, а все *один* (беззаботно).

*Один* у одного на разуме не бывает (не бывали).

Под *один* голосок, под *один* волосок, под одно платище.

Подаришь уехал в Париж, а остался *один* купишь.

Пошло дело на лад: словно *один* держит, другой не пускает.

Развалился, как верста. Распластался *один* на всю лавку.

Семь крутых утренников: три до Власья, *один* на три после Власья.

Сколько (часов) било? — *один* бил, два держил (отвечал немец, которого наказали).

— Служба, не видал ли тут теленочка? — А какой он был? — Рыженький, с обрывком на шее. — А *один* бок потертый? — Да, да, он и есть! — Нет, не видал.

Слушают на гумне: дружный стукоток — идти в большую семью; *один* молотит — за одинокого.

Хозяйка из села Помелова, из деревни Вениковой, пирожок испечет, и корова не ест; да поставит (растворит) три, посадит (в печь) два, а вынет *один*.

Хоть бы уж *один* конец.

Хоть голову с плеч (или: хоть голову ссечь, снести), да *один* бы конец.

Чики, чики, чикалочки, *один* едет на палочке, другой на тележке, щелкает орешки.

Я на всех доспел, а он *один* свертел.

**Прислала Марина Баштанова**

Бог *один*, как ни призывай его.

*Один* кинул — не докинул, другой кинул — перекинул, третий кинул — не попал.

Бредет и зауряд в *один* ряд.

*Один* рядовой, да и тот кривой.

**Привела Анна Волынцева**

*Один* язык перемелется, другой переболтается.

Коли *один* говорит, так двое глядят да двое слушают.

*Один* ура кричит, другой караул.

Дарило умер, остался *один* Купило.

У всякого свой вкус: *один* кладет в карман, другой — в картуз.

У всякого свой вкус: *один* любит арбуз, а другой — свиной хрящик.

Умен, как поп Семен: книги продал, да карты купил, забился в овин да играет *один*.

Лучше *один* раз попробовать горчичное зернышко, чем смотреть на слона, нагруженного горчицей.

**Привела Олеся Исакова**

Лучше дважды спросить, чем *один* раз напугать.

*Один* добытый опыт ценнее семи мудрых поучений.

У волокитчика *один* ответ: приходи завтра, когда меня нет.

Больше слушай, поменьше говори, не зря у тебя два уха и *один* только язык.

Чем десять раз читать, лучше *один* раз записать. (Японская пословица)

Передовик не тот, кто *один* впереди идет, а тот, кто сам шагает и другим помогает.

*Один* день дорог для того, кто делает все вовремя.

*Один* день заменяет три тому, кто умеет все сделать вовремя. (Китайская пословица)

Лучше *один* день подумать, чем целую неделю впустую трудиться. (Финская пословица)

Лучше *один* день быть человеком, чем тысячу дней быть тенью. (Китайская пословица)

*Один* трус — всему войску гибель. (Азербайджанская пословица)

Захворать — *один* день, а поправляться семь недель.

*Один* утренний час лучше двух вечерних. (Английская пословица)

Упал с дерева *один* лист — жди осени. (Японская пословица)

*Один* жил — в заплатках ходил, в колхоз пришел — кафтан нашел.

*Один* раз смеяться — хорошо, много смеяться — похоже на крик осла. (Афганская пословица)

**Прислал Станислав Кулешов**

В дружном табуне *один* хромоу кулан не заметен.

Врагу понятен *один* язык — русская пуля и русский штык.

Все за одного, а *один* за всех — тогда и на фронте будет успех.

Всем стоять как *один* человек.

Выбор всегда *один*: свобода или смерть.

Десять пальцев сильнее, чем *один*.

Друг — *один*, врагов — много.

Друзей, с которыми можно делиться тайной, — много, но сохранять ее может только *один*.

Если в бой — иди *один*.

Если в строю *один* идешь в ногу — смени ее.

Когда объединяется тысяча, *один* рот молчит.

Коль с тобою карабин, значит, ты не *один*.

Кто *один* — тот связан, кто со всеми — тот волен.

Лучше *один* приятель, чем десять врагов.

Лучше сто друзей, чем *один* враг.

*Один* — не ратник в поле, и дерево одно — не целый лес.

*Один* болтает, а другой на ус мотает.

*Один* в лес, а другой на мельницу.

*Один* враг может сделать больше зла, чем сто друзей добра.

*Один* дом джигита — девичья юрта, другой дом — поле битвы.

*Один* дурной поступок порождает другой.

*Один* и у каши не спор.

*Один* идет навстречу опасности, другой прячется в кусты.

*Один* испортит славу тысяч.

*Один* камень не крепость.

*Один* конь пыли не взметет, а если и взметет пыль, то не добудет славы.

*Один* на семерых пойдет.

*Один* патрон в ружье дороже тысячи в поле.

*Один* раз не сумеешь, во второй раз не научишься.

*Один* с огнем, а другой — с пламенем.

*Один* трус — всему войску гибель.

*Один* хороший опыт важнее семи мудрых советов.

*Один* человек не может быть сильным.

*Один* честный покойник лучше ста живущих в позоре.

От паники до измены — *один* шаг.

От трусости до предательства — *один* шаг.

Приказ *один*, а исполнители его не одинаковые.

С другом расстанешься — *один* год будешь плакать; с семьей разлучишься — десять лет будешь плакать; от народа отделишься — весь век будешь плакать.

Слушай тысячу раз, а говори *один* раз.

Смерть бывает *один* раз.

Собравши пальцы в *один* кулак, можно крепко ударить.

Созданное в течение ста лет может быть разрушено за *один* час.

Танкист что артист — и тот и другой должны знать искусство; *один* на сцене стоя, а другой на поле боя.

У народа *один* дом — Родина.

Храбрый умирает *один* раз, трус — сто.

Чего не сможет *один*, то исполнит весь коллектив.

Привела Анна Монахова

Как ни ширься, а *один* всей лавки не займешь.

Мастер *один*, а подносчиков десять.

*Один* не одолеешь и кочку.

*Один* и камень не сдвинешь, а с другом и гору поднимешь.

Все люди как люди, *один* черт в колпаке.

*Один* получил вершки, другой — корешки.

От добра до худа *один* шагжок.

Привела Светлана Пирогова

На *один* рубль долгу три полтины росту.

*Один* “авось” да “как-нибудь” до добра не доведут.

Богатство *один* разум рождает.

*Один* волк пирует, а весь мир горюет.

Как ни кинь, все *один* клин.

Всякий разумен по-своему: *один* спешит, а другой — потом.

*Один* бежит, другой лежит, третий кланяется.

*Один* баранец пасет тысячу овец.

Женка за три угла хату держит, а мужик — за *один*.

Прислала Ольга Ситало

*Один* Бог, одна правда.

Тот и господин, кто все может сделать *один*.

*Один* день год кормит.

*Один* скачешь, *один* и плачешь, а все *один*.

В скотине *один* пар, а души нет.

*Один* в одного не приходится.

*Один* одного краше.

*Один* одного хитрее.

Муж да жена одна плоть, *один* дух.

*Один* как перст, без друга.

*Один*, что верста в поле.

Он у меня *один*, как синь порох в глазу.

*Один*, как солнце, как месяц на небе.

*Один*, что голова на плечах.

*Один*, как петух воевода.

*Один*, как черт в болоте.

Привела Елизавета Стафеева

*Один* шьет, другой порет.

*Один* сын — не сын, наряжать им не кем.

Как не ширься, *один* всей лавки не займешь.

Прислала Анна Шалагина

Окончание в следующем выпуске

## Учителю информатики: памятные даты января

Окончание. См. с. 1–2

выполняемых на ней операций, в частности, появились операции “Печать без сложения”, “Сложение без печати”, “Поперечное сложение”, “Печать списков и таблиц”. Впоследствии приводной рычаг был заменен электрическим двигателем.

В январе 1886 года Бэрроуз и три его компаньона создали Американскую компанию арифмометров — одну из первых в мире фирм, занимавшихся производством счетных машин.

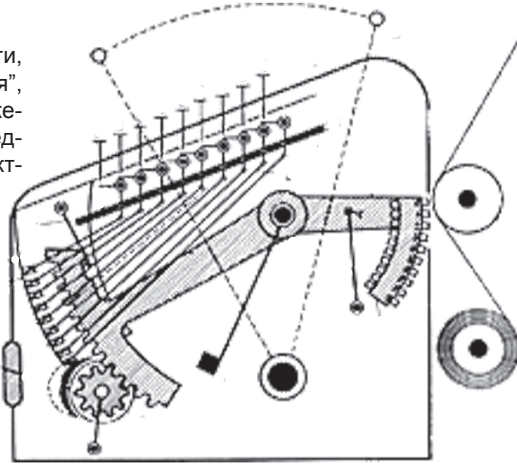
В XX веке корпорация “Бэрроуз” (Burroughs) стала одним из крупнейших в мире производителей компьютеров. В частности, этой фирмой выпущен первый компьютер на интегральных схемах (1968 г.), созданы многопроцессорные системы B5500 (1962 г.), B6500, B6700, B7700, B-3800 (1976 г.), B-4800 (1976 г.), B-6800 (1977 г.), B-7800 (1978 г.), где были воплощены многие новые идеи параллельной обработки данных, и система B-7900 (1982–1983 гг.), в основу архитектуры которой положен принцип так называемой “распределенной обработки” (предполагающий использование нескольких специализированных функциональных подсистем различного назначения и разной производительности) [6].

В 1986 году корпорация Burroughs слилась с корпорацией Sperry Rand, в результате чего образовалась компания Unisys.

На могиле Уильяма Бэрроуза написано [5]: “Здесь покоится человек, который был благородным в бедности, скромным в богатстве и великим в своих делах на благо человечества”.

В январе 1944 года один из создателей первого успешно функционировавшего электронного цифрового компьютера (ENIAC) Джон Эккерт выдвинул идею хранимой программы.

Суть этой революционной для компьютерной техники идеи в том, что “программы ЭВМ должны храниться в ее внутренней памяти наравне с исходными данными и



Машина Бэрроуза

промежуточными результатами вычислений и представляться числовым кодом, что позволяет выполнять над ними те же операции, что и над числами” [5].

### Литература

1. Храмов Ю.А. Физики. Биографический справочник. Изд. 2-е. М.: Наука, Гл. редакция физико-математической литературы, 1983.
2. Стройк Д.Я. Краткий очерк истории математики: Пер. с нем. Изд. 4-е. М.: Наука, 1984.
3. Смирнов О.А., Майорова Т.С., Власова И.Г. 100 великих имен в математике, физике и географии. М.: Филологическое общество “СЛОВО”, 1998.
4. Колмогоров А.Н. Гильберт // Большая советская энциклопедия. Изд. 3-е. Т. 6. М.: Советская энциклопедия, 1973.
5. Полунов Ю.Л. От абака до компьютера: судьбы людей и машин. Книга для чтения по истории вычислительной техники в двух томах. Т. I. М.: Издательско-торговый дом “Русская Редакция”, 2004.
6. Частиков А.П. От калькулятора до суперЭВМ // Новое в жизни, науке, технике. Сер. “Вычислительная техника и ее применение” № 1/1988.

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

ГАЗЕТЫ  
ИЗДАТЕЛЬСКОГО ДОМА  
Первое сентября  
гл. ред. — Е.В. Бирюкова,  
индекс подписки — 32025;

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

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

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

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

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

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

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

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

Французский язык  
гл. ред. — Г.А. Чеснобичкая,  
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  
вклады “В мир информатики”)  
Л.Н. Картвелишвили  
С.Б. Кишкина  
Н.П. Медведева  
Ю.А. Первин (редактор вклады  
“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. 249-48-96  
Отдел рекламы: 249-98-70

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 6000 экз.  
Срок подписания в печать по графику 22.12.2005.  
Номер подписан 22.12.2005.  
Заказ № 615502  
Цена свободная

ИНДЕКС ПОДПИСКИ  
для индивидуальных подписчиков 32291  
комплекта изданий 32744

Тел.: (095) 249-31-38, 249-33-86. Факс (095)249-31-84

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