

Учителю информатики: памятные даты февраля

4 февраля 1943 года в Новом Орлеане (штат Луизиана) родился Кеннет Томпсон — один из создателей операционной системы UNIX, за разработку и реализацию которой в 1983 году ему и Деннису Ричи была вручена премия Тьюринга (Turing Award).

5 февраля 1943 года в городе Клэрвилл (штат Юта) родился Нолан Бушнелл — изобретатель компьютерных игр, основатель компании Atari, один из самых успешных бизнесменов знаменитой Кремниевой долины, которого в прежние годы называли философом Игры [1].

7 февраля 1925 года в городе Днепропетровске родился Николай Петрович Брусенцов — создатель «чудо-машины» — троичного компьютера «Сетунь».



Николай Петрович
Брусенцов

Среди всего многообразия вычислительных машин, построенных в Советском Союзе, существовала машина, отличавшаяся от всех остальных не только техническими характеристиками, но и принципом работы. Это — троичная ЭВМ «Сетунь», созданная в МГУ им. М.В. Ломоносова коллективом под руководством Н.П. Брусенцова (которая производилась серийно с начала до середины 1960-х годов). Она взяла название протекающей вблизи университета реки и относилась к машинам второго поколения, причем была построена на неполупроводниковой элементной базе, а именно — на магнитных элементах.

«...Достоинством «Сетуни» являлось то, что все описание помещалось в маленькие книжечки, причем доступные «простым смертным». Даже гуманитарии пользовались «Сетунию» самостоятельно». В 1961–1968 гг. Брусенцов вместе с Е.А. Жоголевым разработал архитектуру новой машины, названной затем «Сетунь-70»: трудоемкость программ уменьшилась в 5–10 раз при небывалой надежности, понятности, модифицируемости и т.п., а также компактности и скорости. На ее основе впоследствии создали «Наставник» — замечательную систему обучения с помощью компьютера. Несмотря на всеобщее распространение двоичных машин, Н.П. Брусенцов утверждает: «Я уверен, что полноценная информатика не может ограничиться общепринятой сегодня по техническим причинам двоичной системой — основа должна быть троичной. У меня налицо убедительные доказательства верности открытого пути. С какой легкостью была сделана «Сетунь», как просто ее осваивали и продуктивно применяли во всех областях <...>» [2, 3].

7 февраля 1920 года родился Ванг Ан (Wang An) — американский инженер-электронщик и предприниматель китайского происхождения; в 1948 году он изобрел запатентованное устройство на магнитных сердечниках, которое применялось в компьютерах до появления микросхем. В 1951 году Ванг основал компанию Wang Laboratories, которая производила настольные калькуляторы и офисные компьютеры. Фирма Ванга стала одним из наиболее успешных американских высокотехнологичных предприятий 1970-х и начала 1980-х годов. Компанией Ванга разработаны некоторые основные принципы обработки текстов и получено около 40 патентов.

8 февраля 1927 года в Ленинграде родился Николай Яковлевич Матюхин (1927–1984) — главный конструктор вычислительных машин для систем ПВО нашей страны.

Николай Яковлевич Матюхин был главным конструктором многих вычислительных машин и систем, имеющих важное оборонное значение.

А началась карьера Николая Яковлевича в самом начале 1950-х годов, когда он, недавний выпускник Московского энергетического института, фактически возглавил группу разработчиков вычислительной машины М-1, введенной в эксплуатацию уже в 1952 году (то есть практически в то

Продолжение на с. 2

СОДЕРЖАНИЕ НОМЕРА

ОФИЦИАЛЬНЫЕ ДОКУМЕНТЫ

Учебники на будущий учебный год 3–4

ПРЕДЛАГАЮ КОЛЛЕГАМ

А.И. Сенокосов. Борьбсья и искать, найти! 5–9
О.А. Житкова,
Т.И. Панфилова.
VBA в приложении к Excel, Word и Power Point 10–15

ЗАДАЧИ

С.Н. Поздняков,
Д.О. Степуленок.
“Автомат для голосования”.
Задача конкурса “КИО-2005” 16–17

ИСТОРИЯ ИНФОРМАТИКИ

Л.Н. Картвелишвили.
Больше, чем долина 18–19

ПОРТРЕТНАЯ ГАЛЕРЕЯ

Томас Курц 20
Джон Кемени 21

ИНФОРМАЦИЯ

“Рождественским чтениям” — 10 лет 22

СЕМИНАР

С.Б. Гашков.
Системы счисления и их применения 23–30

“НАЧАЛКА” № 3

Газета-клуб для всех, кто учит информатике маленьких детей

Н.Д. Шумилина.
Исполнитель Автомат — так ли все просто? .. 31–34

О.Ю. Кондратьева,
Н.Г. Жукова.
Компьютерная компетентность в контексте развития личности ребенка (из опыта Филимоновской школы) 34–36

“В МИР ИНФОРМАТИКИ”

№ 68
Газета для пытливых учеников и их талантливых учителей

Семинар
Н.М. Тимофеева.
Диаграммы Насси — Шнейдермана 37–40

“Ломаем” голову
Кто изображен на портрете? 40

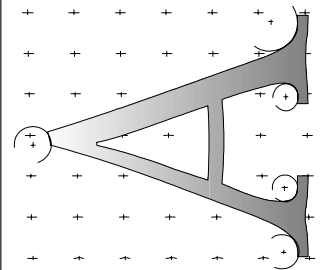
Задачник
В магазине 40

Моделирование
Блуждания “случайного” жука 41–42

Внимание! Конкурс
Итоги конкурса № 40 для учащихся 42–43
Итоги первого тура конкурса № 41 43–47
Конкурс № 43 для учащихся 47

№ 3 (508)

1–15 февраля 2006



Методическая газета для учителей информатики

ИНФОРМАТИК

Учителю информатики: памятные даты февраля

Продолжение. См. с. 1

же время, что и первая отечественная электронная цифровая вычислительная машина МЭСМ (малая электронная счетная машина), созданная под руководством С.А. Лебедева). В М-1, наверное, впервые в мировой вычислительной практике логические схемы были реализованы на полупроводниковой элементной базе, а точнее, на полупроводниковых диодах [1].

11 февраля 1847 года в небольшом городке Майлане (штат Огайо, США) родился Томас Алва Эдисон (1847–1931) — один из самых плодовитых изобретателей в истории человечества.

Эдисон получил более 1000 патентов на свои изобретения [4]. Он усовершенствовал телеграф и телефон, изобрел фонограф (1877). Причина необычайного успеха этого человека прежде всего в том, что он почти никогда не работал над ненужными вещами. Поэтому все его изобретения приходились как нельзя кстати. Кроме того, Эдисон был невероятно трудоспособен. Его рабочий день часто длился 18–20 часов. Однако чрезвычайно важно здесь то, что работал он не один. Лаборатории Эдисона были, по сути дела, научно-исследовательским институтом с десятками специалистов (а для воплощения в жизнь своих идей Эдисон создавал целые заводы). Многие из них потом сами стали выдающимися изобретателями [5].

11 февраля 1915 года в Чикаго родился Ричард Весли Хэмминг (1915–1998) — один из основоположников теории кодирования, автор знаменитого учебника “Теория кодирования и теория информации”, переведенного в 1983 году на русский язык.

11 февраля 1927 года в Москве родился Всеволод Сергеевич Бурцев (1927–2005) — академик, один из крупнейших в нашей стране конструкторов высокопроизводительных вычислительных машин и комплексов для управления объектами, работающими в масштабе реального времени. В.С. Бурцеву принадлежат такие слова (2003 г.): “За рубежом, в частности в США, идут по пути создания многопроцессорных суперкомпьютеров. Такие машины содержат тысячи процессоров. Теоретическая (максимальная) производительность растёт пропорционально числу процессоров, а реальная производительность процессоров падает. В своих работах мы показываем, что это тупиковое направление. За американцами по нему следовать нет резона. Нужно делать процессоры “массового параллелизма” с новой архитектурой...” [6].



Всеволод Сергеевич Бурцев

В 1973–1985 годах Всеволод Сергеевич руководил разработкой суперкомпьютеров “Эльбрус-1” (с производительностью до 15 млн. оп/с) и “Эльбрус-2” (с производительностью до 125 млн. оп/с). Для обеих систем удалось наладить серийное производство.

Немного позже под руководством Бурцева была разработана архитектура суперЭВМ, основанная на новом, “не фоннеймановском” принципе, которая обеспечивает распараллеливание вычислительного процесса на аппаратном уровне [1]. Принципиальная особенность предложенной архитектуры — автоматическое динамическое распределение вычислительных ресурсов между отдельными процессами (что призвано освободить человека от необходимости распределения ресурсов при программи-

ровании параллельных процессов для многомашинных и многопроцессорных комплексов).

12 февраля 1973 года в Торонто началось возведение самой высокой постройки мира — Канадиен Нэшнл Тауэр (Canadian National Tower), или сокращенно К.Н. Тауэр, высота которой вместе со шпилем антенны составляет 553,5 м. Многие века самым высоким сооружением в мире являлась пирамида Хеопса, чья первоначальная высота составляла 146,59 м. Когда в 1886 году инженер Гюстав Эйфель представил чертежи своей башни в главный совет международной выставки, самым высоким сооружением в мире был Кельнский собор, который вместе со шпилем вытянулся на 156 м. Эйфель же предлагал построить сооружение в два раза выше. С 1889-го по 1931 г. Эйфелева башня оставалась самой высокой постройкой, с 1931 г. это звание носил небоскреб Эмпайр-стейт-билдинг, а затем пальма первенства перешла к Останкинской телебашне в Москве высотой 537 м.

В 1975 г. рекорд побила К.Н. Тауэр, которую строили 40 месяцев. Идея заключалась в том, чтобы башня могла принимать и передавать сигналы, а при этом многочисленные небоскребы Торонто не создавали ей помех. Кроме своего практического назначения, башня стала еще и местом развлечений, достопримечательностью для туристов, а главное — символом Торонто [7].

13 февраля 1805 года в Дюрене (Рейнская провинция) родился Петер Густав Лежён Дирихле (1805–1859) — крупнейший немецкий математик, чье творчество, связанное в основном с теорией чисел, теорией рядов, интегральным исчислением и некоторыми проблемами математической физики, существенно повлияло на развитие математических знаний.

13 февраля 1910 года в Лондоне родился Уильям Брэдфорд Шокли (1910–1989) — американский физик, один из изобретателей транзистора (полупроводникового триода).

“Примечательно, что изобретатели натолкнулись на возможность создания полупроводниковых приборов довольно давно: в частности, еще... нижегородский радиоинженер О.Лосев [Олег Владимирович Лосев (1903–1942). — Прим. ред.] создал (в 1920-е гг. — Прим. ред.) первые полупроводниковые усилители и генераторы. О его работах был широко оповещен мир, у крупнейших радиотехнических лабораторий появилась возможность развивать полупроводниковую технику. Однако процесс этот задержался почти на два десятилетия: еще не было в то время фундамента для становления полупроводниковой электроники. Она стала возможной лишь после того, как были детально изучены и глубоко поняты тонкие и сложные физические процессы в твердом теле”. (Академик Е.П. Велихов, [14]). В современной электронике транзистор — основной элемент, а интегральная микросхема (наиболее значительное достижение микроэлектронной технологии) представляет собой кристалл полупроводника, в объеме и на поверхности которого сформированы, например, транзисторы и соединения между ними.

Когда появились первые транзисторы, мало кто мог оценить их надлежащим образом, однако позже транзистор признали одним из важнейших изобретений XX века, которое позволило автоматизировать многие процессы и способствовало коренному изменению жизни людей [8].

В 1956 году троим ученым научно-исследовательской фирмы Bell Telephone Laboratories, входящей в американскую компанию AT&T (American Telephone and Telegraph), Уильяму Брэдфорду Шокли, Джону Бардину и Уолтеру Браттейну, была присуждена Нобелевская премия по физике “за исследования полупроводников и открытие транзисторного эффекта”.

Окончание на с. 48

Учебники на будущий учебный год

Извлечение из федерального перечня учебников, рекомендованных (допущенных) Министерством образования и науки Российской Федерации к использованию в образовательном процессе в общеобразовательных учреждениях, на 2006/2007 учебный год

Настоящий перечень утвержден Приказом Минобрнауки России № 302 от 07.12.2005. Учебники по информатике содержатся только в первой части перечня (в Приложении 1). Перечень будет опубликован в официальном издании Минобрнауки России — сборнике приказов и инструкций Министерства образования и науки Российской Федерации “Вестник образования”. Электронная версия федеральных перечней размещена на сайте Минобрнауки России (www.mon.gov.ru).

Для удобства ссылок на позиции перечня в публикуемом извлечении мы сохранили номера пунктов, соответствующих учебникам по информатике.

№ п/п	Автор (составитель), название, количество частей, класс	Год присвоения грифа	Издательство	Дополнительная информация издательства
Раздел I. НАЧАЛЬНОЕ ОБЩЕЕ ОБРАЗОВАНИЕ				
Информатика				
305	Бененсон Е.П., Паутова А.Г. Информатика. Ч. 1, 2. 2 кл.	2003	Академкнига / Учебник	Комплект “Перспективная начальная школа”. Дополнительно: методическое пособие для учителя (Бененсон Е.П., Паутова А.Г.), программа
306	Горячев А.В. и др. Информатика в играх и задачах. Учебник-тетрадь. Ч. 1, 2. 1 кл.	2005	Баласс	Образовательная система “Школа-2100” под ред. Леонтьева А.А. Методические рекомендации, наглядные пособия
307	Горячев А.В. и др. Информатика в играх и задачах. Учебник-тетрадь. Ч. 1, 2. 2 кл.	2005	Баласс	Образовательная система “Школа-2100” под ред. Леонтьева А.А. Методические рекомендации, наглядные пособия
308	Горячев А.В. и др. Информатика в играх и задачах. Учебник-тетрадь. Ч. 1, 2. 3 кл.	2005	Баласс	Образовательная система “Школа-2100” под ред. Леонтьева А.А. Методические рекомендации, наглядные пособия
309	Горячев А.В. и др. Информатика в играх и задачах. Учебник-тетрадь. Ч. 1, 2. 4 кл.	2005	Баласс	Образовательная система “Школа-2100” под ред. Леонтьева А.А. Методические рекомендации, наглядные пособия
310	Матвеева Н.В. и др. Информатика. 2 кл.	2003	БИНОМ	УМК: учебник 2 кл.; учебник 3 кл.; рабочие тетради 2 кл. № 1, 2; 3 кл. № 1, 2; контрольные работы 3 кл. Дополнительно: плакаты 2–4 кл.
311	Тур С.Н., Бокучава Т.П. Информатика. Учебник-тетрадь. 2 кл.	2005	БХВ-Петербург	Учебники-тетради для 2, 3, 4 кл. являются частью непрерывного курса информатики с 1 по 9 кл. Учебники-тетради сопровождаются методическим пособием для учителей 2–4 классов и компакт-диск с пакетом педагогических программных средств “Страна Фантазия” и программой курса. Учебник-тетрадь для 4 кл. имеет вкладыш для контрольных и самостоятельных работ
312	Тур С.Н., Бокучава Т.П. Информатика. Учебник-тетрадь. 3 кл.	2005	БХВ-Петербург	
313	Тур С.Н., Бокучава Т.П. Информатика. Учебник-тетрадь. 4 кл.	2005	БХВ-Петербург	

№ п/п	Автор (составитель), название, количество частей, класс	Год присвоения грифа	Издательство	Дополнительная информация издательства
Раздел II. ОСНОВНОЕ ОБЩЕЕ ОБРАЗОВАНИЕ				
Информатика и ИКТ				
532	Босова Л.А. Информатика. 5 кл.	2003	БИНОМ	УМК 5–6 кл.: учебник 5 кл.; рабочая тетрадь 5 кл.; учебник 6 кл.; рабочая тетрадь 6 кл.; задачник 5–6 кл.; методическое пособие 5–6 кл. Дополнительно: плакаты 5–6 кл.
533	Босова Л.А. Информатика. 6 кл.	2004	БИНОМ	
534	Угринович Н.Д. Информатика и ИКТ. 7 кл.	2004	БИНОМ	Соответствуют федеральному компоненту стандарта 2004 г. УМК 7–9 кл. (8–9 кл.): учебник 7 кл.; учебник 8 кл.; учебник 9 кл.; методическое пособие 7–11 кл. Дополнительно: плакаты 7–9 кл.
535	Угринович Н.Д. Информатика и ИКТ. 8 кл.	2004	БИНОМ	
536	Угринович Н.Д. Информатика и ИКТ. 9 кл.	2004	БИНОМ	
537	Семакин И.Г. и др. Информатика и ИКТ. 8 кл.	2004	БИНОМ	Соответствуют федеральному компоненту стандарта 2004 г. УМК 8–9 кл.: учебники 8 кл.; 9 кл.; задачник-практикум 8–11 кл. (в двух томах); структурированный
538	Семакин И.Г. и др. Информатика и ИКТ. 9 кл.	2004	БИНОМ	Конспект базового курса; методическое пособие 8–9 кл. Дополнительно: плакаты 7–9 кл.
539	Под ред. Макаровой Н.В. Информатика. Начальный курс. 5–6 кл.	2003	Питер Принт	Учебник является частью учебно-методического комплекта по информатике 5–11 кл. и охватывает пропедевтический этап изучения информатики. В комплект входят: “Рабочая тетрадь по информатике. 5 кл.”; “Рабочая тетрадь по информатике. 6 кл.”
540	Под ред. Макаровой Н.В. Информатика. Базовый курс. 7–9 кл.	2001	Питер Принт	Является частью учебно-методического комплекта по информатике 5–11 кл. В комплекте: практикум по информационным технологиям, практикум-задачник по моделированию, методическое пособие для учителей, программа по информатике
Раздел III. СРЕДНЕЕ (ПОЛНОЕ) ОБЩЕЕ ОБРАЗОВАНИЕ				
Информатика				
876	Угринович Н.Д. Информатика и информационные технологии. 10–11 кл.	2002	БИНОМ	Соответствует обязательному минимуму содержания образования 1999 г. УМК: учебник 10–11 кл., практикум, методическое пособие 7–11 кл.: “Исследование информационных моделей”; “Основы программирования на примере Visual Basic .NET”
877	Семакин И.Г., Хеннер Е.К. Информатика. 10 кл.	2001	БИНОМ	Соответствует обязательному минимуму содержания образования 1999 г. УМК: учебник 10 кл., учебник 11 кл., задачник-практикум 8–11 кл. (в двух томах), элективный курс “Информационные системы и модели”
878	Семакин И.Г., Хеннер Е.К. Информатика. 11 кл.	2002	БИНОМ	
879	Бешенков С.А., Ракитина Е.А. Информатика. 10 кл.	2001	БИНОМ	Соответствует обязательному минимуму содержания образования 1999 г. УМК: учебник 10 кл., учебник 11 кл.
880	Бешенков С.А., Кузьмина Н.В., Ракитина Е.А. Информатика. 11 кл.	2002	БИНОМ	

Бороться и искать, найти!

А.И. СЕНОКОСОВ,

г. Екатеринбург

Часть 1. Поиск текстов в Интернете. Основные понятия

Для начала напомним, что история WWW (World Wide Web — Мировой информационной паутины) началась в марте 1989 года, когда Тим Бернерс-Ли из Европейской лаборатории физики элементарных частиц (CERN) предложил новый способ обмена результатами исследований и идеями между участниками коллектива исследователей-физиков, работавших в разных странах. Для передачи документов и установления связи предлагалось использовать просто систему гипертекста (тогда никто еще не задумывался о возможности передачи фотографических изображений, звука или видео; речь шла только о распространении текстовых документов, содержащих гиперссылки на фрагменты других таких же текстовых документов, но располагающихся на удаленных компьютерах, подключенных к глобальной сети Интернет).

Собственно, гипертекст не был изобретением Бернерса-Ли. Любой, кто пользовался справочной системой какой-либо программы, это подтвердит. Да и HTML вовсе не был первым языком описания страниц. Но в нем впервые были связаны воедино не просто отдельные главы одного документа, а документы, располагающиеся на самых различных серверах Интернета по всему земному шару.

До конца 1992 года программа оставалась лишь частью академических исследований — демонстрировалась на семинарах в CERN'e, да еще была представлена на конференции "Гипертекст'91".

Переломный момент наступил в декабре 1992 г.: Марк Андрессен вместе с Эриком Бина задумали написать свою собственную программу-клиент для просмотра гипертекстовых WWW-документов. И в марте следующего года (всего через три месяца!) на свет появилась Mosaic — мощная программа с графическим интерфейсом для работы с гипертекстовыми страницами и другими ресурсами Интернета. С этого момента пользователь уже мог не вдаваться в технические подробности реализации протоколов компьютерных сетей, а, используя очень простой и удобный инструмент, посвятить всего себя работе с интересующей его информацией, распределенной по огромному множеству компьютеров, расположенных в разных городах, странах и даже на разных континентах.

Mosaic в кратчайший срок стала самой быстро распространяющейся по миру программой. Таким образом, недавний выпускник колледжа Марк Андрессен добавил последний штрих, позволивший превратить всю Сеть, насчитывавшую в то время от одного до двух миллионов компьютеров, в единый гигантский супер-

источник информации, работа с которым ничуть не труднее, чем для 10-летнего ребенка запуск любимой игры на домашнем персональном компьютере.

С каждым месяцем, днем и часом в гигантской паутине гипертекстовых связей появляются все новые и новые мега- и гигабайты информации, и теперь всякий исследователь, прокладываящий себе путь в необъятных просторах киберпространства, с первых же минут ощущает потребность в надежном путеводителе. Дэвид Фило и Джерри Янг из Станфордского университета в апреле 1994 года пришли к выводу, что у каждого из них накопилось такое большое количество ссылок на самые разные информационные источники, что пора каким-то образом их упорядочить.

Так родилась идея создания специализированной базы данных, которая вскоре стала использоваться тысячами пользователей для эффективного поиска информации в Сети. Называется эта поисковая система "Yahoo!". Для широких масс коллекционеров совершенно бесполезной информации сообщаем, что расшифровывается это просто и понятно: "Yet Another Hierarchical Official Oracle" ("Еще один иерархический официальный оракул").

В начале следующего, 1995 года уже упоминавшийся выше Марк Андрессен помог Фило и Янгу переместить их детище на более мощные компьютеры Netscape, и вскоре система Yahoo! стала самым популярным и полным иерархическим предметно-ориентированным путеводителем по Всемирной паутине и Интернету в целом. В Yahoo! все рассортировано по темам и категориям, вся информация разложена по полочкам так, что вроде бы даже самому ленивому пользователю не составит большого труда найти искомый ресурс в необозримом пространстве Сети...

Действительность, как обычно, оказалась несколько сложнее. И регулярные конкурсы по поиску информации в Интернете — лишнее тому подтверждение. Впрочем, легко убедиться в существовании проблемы, когда в ответ на простое вроде бы словосочетание "закон Ньютона" система выдает список самых разнообразных страниц — от тестирования "Ягуара" до сайтов сомнительного содержания. В общем, далеко не все так гладко, как обычно описывают в тощеньких пособиях из разряда "Все, что вы хотели знать об Интернете, но почему-то боялись спросить".

Итак, когда говорят о поиске информации в Интернете, неявным образом имеют в виду работу с поисковыми машинами.

Поисковые машины — это не просто специальные сайты в глобальной сети, которые помогают найти нужную информацию. Это очень крупные предприятия ИТ-индустрии, на которых работают тысячи самых раз-

личных специалистов. В качестве примера можно привести компанию Google, для которой прошедший 2005 год стал звездным. И дело не только в том, что ей принадлежит один из самых мощных поисковых серверов, но и в постоянном, порой революционном развитии информационных технологий, связанных с Интернетом. Она одна из первых предложила пользователям сотовых телефонов специализированную электронную почту, организовала поистине сенсационный проект Google Earth, давший возможность интернет-пользователям взглянуть на нашу планету с высоты полета спутников (заметим, совершенно бесплатно). Поражает воображение и новый проект Google Books, посвященный оцифровке максимально возможного количества бумажных книг и размещению их в Интернете. Также стоит напомнить, что Google работает над системой языкового перевода (Google Translate), который в перспективе сделает доступным все ресурсы Интернета любому пользователю, на каком бы языке он ни говорил.

Все это позволило не такой уж большой по индустриальным меркам компьютерной компании собрать гигантское количество инвестиций и обогнать по капитализации (стоимости акций) крупнейшую медиакомпанию Time Warner со всем ее Голливудом. Понятно, что авторитетное издание Financial Times особенно долго предварительно не думало, назвав “людьми 2005 года” руководителей Google Сергея Брина и Ларри Пейджа (что, кстати, дополнительно подняло стоимость ее акций более чем на 2%. Теперь рыночная стоимость Google составляет свыше 120 миллиардов долларов, приближаясь к стоимости “голубого гиганта” — IBM (130 миллиардов долларов)).

Итак, разберем поподробнее, как же работает то, с чего, собственно, и начался взлет этой компании, признанного лидера современного Интернета, — поисковая машина.

После того, как вы ввели во входную строку поисковой машины искомое словосочетание, происходит обращение к базе данных сервера, и буквально через пару-другую секунд выдается список web-страниц и сайтов, в которых встречаются заданные слова и/или их сочетание. Иными словами, нужный вам документ или файл уже был когда-то найден, а информация о нем попала в базу данных.

Чтобы найти информацию о сотнях миллионов существующих web-страниц, поисковая машина применяет специальную программу-робот. Эта программа еще называется “спайдер” (*spider* — от англ. “паук”) и служит для построения списка слов, найденных на странице. Процесс построения такого списка называется web-краулинг (*Web crawling* — “ползание по паутине”). Чтобы построить и зафиксировать полезный, т.е. имеющий значение, список слов, поисковый паук должен просмотреть сотни тысяч и миллионы страниц.

Как же начинает программа-паук свое путешествие по сети? Обычно стартовой точкой являются наиболее

крупные мировые серверы и очень популярные web-страницы. Паук начинает свой путь с такого сайта, заносит в базу данных (индексирует) все найденные слова и продолжает свое движение далее — по ссылкам на другие сайты. Таким образом, робот-паук начинает охватывать всё большие куски web-пространства.

Google.com начинался с достаточно стандартной программы-паука. Сергей Брин и Ларри Пейдж рассказали об истории ее создания и общих принципах действия. Обычно поиск в Интернете начинался с использованием трех пауков. Каждый паук поддерживал до 300 одновременно открытых соединений с web-страницами. При пиковой загрузке, с использованием четырех пауков, система Google была способна обработать более 100 страниц в секунду. Чтобы ускорить работу своих “пауков”, компания обзавелась собственным DNS-сервером, сведя до минимума временные затраты на индексацию страниц. (Напомним, что именно DNS-серверы позволяют навести единый порядок в Сети, сопоставляя буквенному названию web-сайта его истинный 16-ричный адрес.)

Разумеется, сейчас мощности системы возросли, но принцип остался неизменным. Итак, когда Google-робот посещает HTML-страницу, он принимает во внимание 2 вещи:

- слова (текст) на странице;
- место их расположения.

Слова, расположенные в служебных тегах-контейнерах, таких, как <TITLE>, <META> и др., помечаются как особо важные для пользовательских поисковых запросов. Google-паук построен так, чтобы индексировать каждое подобное слово на странице, за исключением междометий и артиклей. Вообще разные программы-пауки имеют каждая свой подход к индексации. Впрочем, все подходы и алгоритмы в конечном итоге направлены на то, чтобы заставить роботов-пауков работать быстрее и эффективнее. К примеру, некоторые поисковые роботы отслеживают при индексации не только слова в контейнере <TITLE>, ссылках и до 100 наиболее часто используемых на странице слов, но и каждое из слов первых 20 строк текстового содержания.

Немного подробнее о метатегах. Они позволяют владельцу web-страницы самому задавать ключевые слова и понятия, которые определяют суть ее содержания. Это очень полезный инструмент, особенно когда эти ключевые слова нечасто попадают в тексте страницы. В этом случае метатеги могут направить поисковый робот к нужному выбору ключевых слов для индексации страницы. Разумеется, существует вероятность того, что в метатегах сайта, скажем, мебельного магазина появятся слова “Мария Шарапова”, что резко увеличит его посещаемость, что никак не связано с содержанием самой страницы. Вообще-то поисковые роботы борются с этим путем, например, сравнения метатегов и содержимого web-страницы, но на практике такая борьба далеко не всегда приводит к успеху.

Как только пауки закончили свою работу по нахождению новых web-страниц, поисковые машины должны разместить всю найденную информацию так, чтобы было удобно в дальнейшем ею пользоваться. Здесь имеют значение 2 ключевых фактора:

- информация, сохраненная вместе с данными;
- метод, которым эта информация проиндексирована.

Фактически это задача создания эффективной базы данных, способной отвечать на миллионы запросов в течение одного часа (для примера — нагрузка на отечественную поисковую машину Яндекс составляет в среднем 9–10 миллионов запросов в день).

В простейшем случае можно было бы просто поместить в базу данных слово и URL-адрес, где оно находится. Но это сделало бы поисковик совсем примитивным инструментом, так как нет никакой информации о том, в какой части документа находится это слово (в мета-тегах или в обычном тексте), используется ли это слово один раз или многократно и содержится ли оно в ссылке на другой важный и близкий по теме ресурс. Другими словами, такой способ не позволит ранжировать сайты, не обеспечит представление пользователям результатов, отсортированных по частоте упоминания, и т.д.

Чтобы предоставить нам полезные данные, поисковики сохраняют не только информацию из слова и его URL-адрес. Поисковик может сохранить данные о количестве (частоте) упоминаний слова на странице, присвоить слову вес, что далее поможет выдавать поисковые листинги (результаты) на основе весового ранжирования по данному слову, с учетом его местонахождения (в ссылках, мета-тегах, титуле страницы и т.п.). У каждого коммерческого поисковика есть своя формула для вычисления веса ключевых слов при индексации. Это одна из причин, почему по одному и тому же поисковому запросу поисковики выдают совсем разные результаты.

Разумеется, программа не может ни понять, ни как-либо классифицировать то, что она видит в Интернете. Роботы, как правило, ограничиваются лишь сбором статистической информации и построением словоуказателей, или индексов, по текстам документов. Собираемая роботом база данных — индекс — содержит, попросту говоря, сведения о том, в каких www-документах содержатся те или иные слова.

Именно автоматически собираемый индекс и лежит в основе таких поисковых систем, как Google, Яндекс, Рамблер. Переоценить их значение невозможно — поиск по ключевым словам в одной базе данных, занимающий в худшем случае несколько секунд, принесет вам те же результаты, что и просмотр всех www-страниц во всем Интернете, на что даже самому современному компьютеру требуется не одна неделя. Предпринять же что-либо подобное вручную просто невозможно.

К сожалению, у вас нет никаких гарантий того, что автоматический индекс охватывает весь Интернет. Доказательством этому может служить хотя бы то, что

таких индексов в Интернете немало, и их базы данных не очень-то пересекаются.

Таким образом, автоматический индекс состоит из трех частей: программы-робота, собираемой этим роботом базы данных и интерфейса для поиска в этой базе, с которым и работает пользователь. Все эти компоненты вполне могут функционировать без вмешательства человека, поэтому успех такой системы определяется по большей части удачно написанными программами и хорошим материальным обеспечением — мощными компьютерами и надежной высокоскоростной связью с Интернетом.

Поскольку какая-либо классификация или оценивание материалов в системах такого рода отсутствует, к ним следует прибегать только тогда, когда вы точно знаете ключевые слова, относящиеся к тому, что вам нужно, — скажем, фамилию человека или несколько достаточно редких терминов из соответствующей области. Если же задать поиск по сколько-нибудь распространенным словам, то вам не хватит жизни, чтобы обойти все полученные в результате поиска URL-адреса — к примеру, индекс системы AltaVista содержит 11 миллиардов слов и словосочетаний, извлеченных из 30 миллионов www-страниц.

Возвращаясь к системе Yahoo!, с которой мы начали наш разговор, следует заметить, что принцип ее работы отличается от рассмотренного нами выше принципа автоматического индексирования.

Поисковые инструменты такого типа чаще всего называют предметными, или тематическими, каталогами (subject catalogs). Компания, владеющая таким каталогом, непрерывно ведет огромную работу, исследуя, описывая, каталогизируя и раскладывая по полочкам содержимое www-серверов и других сетевых ресурсов, разбросанных по всему миру. Результатом ее поистине титанических усилий является огромный и постоянно обновляющийся иерархический (древовидный) каталог, на верхнем уровне которого собраны самые общие категории, такие, как “бизнес”, “наука”, “искусство” и т.п., а элементы самого низкого уровня представляют собой ссылки на отдельные www-страницы и серверы вместе с кратким описанием их содержимого.

Гарантий того, что такой каталог действительно охватывает все содержимое WWW, вам, естественно, никто не даст — составляют каталог все-таки люди, а людям свойственно иногда уставать и ошибаться. Однако возможная неполнота и даже однобокость подбора материалов в таких каталогах с лихвой искупается тем, что пока еще не под силу никакому роботу, — *осмысленностью* отбора. Информация, которой коснулась рука человека, становится во всех смыслах дороже — практической пользы от нее больше, но и себестоимость такой информации выше.

Упорядоченность содержимого в тематическом каталоге делает его незаменимым в тех случаях, когда вы изначально не имеете почти никакой информации о цели своих поисков. Например, если вам нужно выяс-

нить, как в мире обстоят дела с производством обуви для езды на велосипеде, вам не обязательно знать какие-либо ключевые слова или названия конкретных фирм: достаточно просто начать спускаться сверху вниз по иерархии (Business and Economy → Companies → Sports → Cycling → Shoes) — и в конце концов вы отыщете раздел, посвященный исключительно велосипедной обуви.

Конечно, при этом вы не застрахованы от попадания в тупик, — и все же рано или поздно вы скорее всего обнаружите именно то, что искали. По содержимому найденного раздела вам сразу станет ясно, какие из фирм этой отрасли представлены в Интернете и какую информацию на эту тему вообще имеет смысл искать в сети.

Предметные каталоги предоставляют и возможность автоматического поиска по ключевым словам. Однако поиск этот происходит не в содержимом самих WWW-серверов, а в их кратких описаниях, хранящихся в каталоге. Иначе говоря, если вы знаете, как называется велосипедная обувь по-английски, вы можете ввести это магическое сочетание слов в специальное поле, нажать кнопку — и сервер сразу сообщит вам, что на эту тему в его иерархии имеется отдельный раздел, и поможет перенестись в него одним щелчком мыши.

Предметные каталоги Интернета можно пересчитать буквально на пальцах, так как их создание и поддержка требуют огромных затрат. Наиболее известные каталоги — Yahoo!, WWW Virtual Library, Galaxy и некоторые другие.

Разумеется, использовать Yahoo! для поиска русскоязычной информации совершенно бесполезно. Но существует и двуязычный (русско-английский) каталог с ненавязчивым адресом *www.ru*. Каталог поддерживается интернет-провайдером компанией “Демос-Интернет”.

Все же чаще всего в повседневной жизни обычный россиянин пользуется автоматическими индексами, и поэтому более подробно рассмотрим такой их важнейший компонент, как интерфейс пользователя. А поскольку он представляет собой не что иное, как строку запроса к базе данных, то и предметом нашего внимания будет специализированный язык запросов, близкий родственник языка SQL, о котором много раз рассказывалось в нашей газете. Способы обработки запросов в различных поисковиках несколько отличаются друг от друга. Поэтому в качестве примера мы рассмотрим работу поискового сервера Яндекс.

Если вы внимательно прочитали предыдущий материал, то наверняка поняли, что хороший запрос — важная предпосылка любого поиска.

Предположим, вы захотели найти в интернет-магазине вашего региона телевизор подешевле. Запрос, состоящий из одного слова — “телевизор”, — это худшее, что может быть. Миллионы ссылок (более конкретно, 17 миллионов), которые вы получите в ответ на это, вряд ли приблизят вас к решению задачи. Разумеется, одними из первых среди этих ссылок будут адреса

крупных торговых сетей, web-программисты которых позаботились о хорошей рекламе. Мало того, сама система Яндекс, зная ваш IP-адрес и, соответственно, сопоставляя с ним регион, предложит интернет-магазин (один-два), обслуживающий его. Но вот что касается более дешевого предложения — можно считать, что мы ни на шаг не продвинулись к достижению цели.

Вообще программное обеспечение поисковиков, анализирующее ваш запрос, за последние годы серьезно улучшилось, и если раньше искали в основном по одному-двум словам, то теперь вполне можно задать вопрос по-человечески. И действительно, запрос “Где купить дешевый телевизор” с отметкой своего региона выдает всего несколько ссылок, из которых парочка — на самом деле то, что мы искали.

Мало того, искомое нами словосочетание скорее всего характерно не для web-страниц, а для форумов и онлайн-дневников (блогов), которые стали в последнее время особенно популярными в Интернете. Система Яндекс предлагает провести поиск и в них. Здесь мы сможем найти ответ на свой вопрос уже из диалога (или монолога) конкретных людей.

Вообще поиск какого-либо товара — это одна из самых простых проблем, поскольку во всех поисковиках предусмотрены соответствующие механизмы. Оно и понятно — не забывайте, что вы пользуетесь поисковой машиной совершенно бесплатно, и коммерчески выгодным предприятием поисковик делает именно реклама. Как уже говорилось выше, можно сузить поиск по категории каталога или по региону. Здесь же указывается, есть ли последние новости по данному запросу и есть ли соответствующие ему товары в интернет-магазинах. Яндекс также обнаруживает и предлагает исправить опечатки и ошибки набора.

Если в результате поиска было найдено слишком много документов, а вы хотели бы сократить этот список, можно поступить двумя способами.

Во-первых, есть возможность “искать в найденном” — для этого надо поставить флажок в форме поиска. Тогда следующий поиск будет вестись не по всем документам, а только по тем, которые были найдены в предыдущем поиске. Например, чтобы найти медицинские страницы про “рак” (а не астрологические), следует задать запрос “медицина онкология”, для определения области поиска. А затем, поставив галочку “искать в найденном”, — слово “рак”.

Во-вторых, можно ограничить область поиска разделом каталога или регионом (в полосе под поисковой формой). Нажатие на ссылку (название темы) задает поиск только по страницам сайтов, относящихся к этой теме.

По умолчанию результаты поиска отсортированы по релевантности (частоте нахождения данного слова или словосочетания) и сгруппированы по серверам. Найденные документы могут иметь пометки “совпадение фразы”, “строгое соответствие” или “нестрогое соответствие” (по мере убывания релевантности). “Не-

строгие” документы отделены от прочих горизонтальной серой линией. Если первым же найденным оказывается документ с подписью “нестрогое соответствие”, возможно, это значит, что стоит переформулировать запрос.

Если в списке найденного нажать на ссылку “Показать найденные слова”, будет показана так называемая “подсветка”. В открытый документ добавляются две маленькие таблички. В верхней находится ссылка на “Первое вхождение”, по нажатию на которую вы попадаете на первое найденное в этом документе слово. Слово обрамляется стрелочками, которые являются, соответственно, ссылками на предыдущее и последующее найденное слово. В нижней табличке содержится статистика найденного, а также сообщение в случае, если документ был изменен после индексации.

Если один из найденных документов ближе к искомой теме, чем остальные, нажмите на ссылку “найти похожие документы” (она расположена под кратким описанием этого документа). Яндекс проанализирует страницу и найдет документы, похожие на указанный вами. Но если эта страница была удалена с сервера и ее нет в памяти Яндекса, то вы получите сообщение “Запрошенный документ не найден”.

В конце каждой страницы с результатами поиска находятся ссылки “Поискать то же самое на: AltaVista — Google — Fast — Yahoo! — Rambler — Апорт!”.

Возможно, такая ссылка будет полезна, если поисковый запрос не принес желаемых результатов. Ваш запрос будет передан в другие поисковые системы и каталоги. В случае поиска на иностранном языке будет не лишним передать запрос в Google или AltaVista.

Советы по поиску в Яндексе

Проверяйте орфографию

Если поиск не нашел ни одного документа, то вы, возможно, допустили орфографическую ошибку в написании слова. Проверьте правильность написания. Если вы использовали при поиске несколько слов, то посмотрите на количество каждого из слов в найденных документах (перед их списком после фразы “Результат поиска”). Какое-то из слов не встречается ни разу? Скорее всего его вы и написали неверно.

Используйте синонимы

Если список найденных страниц слишком мал или не содержит полезных страниц, попробуйте изменить слово. Например, вместо “рефераты”, возможно, больше подойдет “курсовые работы” или “сочинения”. Попробуйте задать для поиска три-четыре слова-синонима сразу. Для этого перечислите их через вертикальную черту (|). Тогда будут найдены страницы, где встречается хотя бы одно из них. Например, вместо “фотографии” попробуйте “фотографии | фото | фотоснимки”.

Ищите больше, чем по одному слову

Слово “психология” или “продукты” дадут при поиске поодиночке большое число бессмысленных ссы-

лок. Добавьте одно или два ключевых слова, связанных с искомой темой. Например, “психология Юнга” или “продажа и покупка продовольствия”. Рекомендуем также сужать область вашего вопроса. Если вы интересуетесь автомобилями ГАЗа, то запросы “автомобиль Волга” или “автомобиль ГАЗ” выдадут более подходящие документы, чем “легковые автомобили”.

Не пишите большими буквами

Начиная слово с большой буквы, вы не найдете слов, написанных с маленькой буквы, если это слово не первое в предложении. Поэтому не набирайте обычные слова с Большой Буквы, даже если с них начинается ваш вопрос Яндексу. Заглавные буквы в запросе рекомендуется использовать только в именах собственных. Например, “группа Черный кофе”, “телепередача Здоровье”.

Ищите похожие документы

Если один из найденных документов ближе к искомой теме, чем остальные, нажмите на ссылку “найти похожие документы”. Ссылка расположена под краткими описаниями найденных документов. Яндекс проанализирует страницу и найдет документы, похожие на тот, что вы указали. Но если эта страница была стерта с сервера, а Яндекс еще не успел удалить ее из базы, то вы получите сообщение “Запрошенный документ не найден”.

Используйте знаки “+” и “-”

Чтобы исключить документы, где встречается определенное слово, поставьте перед ним знак минус. И наоборот, чтобы определенное слово обязательно присутствовало в документе, поставьте перед ним плюс. Обратите внимание, что между словом и знаком плюс-минус не должно быть пробела. Например, если вам нужно описание Парижа, а не предложения многочисленных турагентств, имеет смысл задать такой запрос “путеводитель по Парижу -агентство -тур”. Плюс стоит использовать в том случае, когда нужно найти так называемые “стоп-слова” (наиболее частотные слова русского языка, в основном это местоимения, предлоги, частицы). Чтобы найти цитату из Гамлета, надо задать запрос “+быть или +не быть”.

Кроме того, при поиске можно использовать и язык запросов, но об этом — в следующем раз.

Использованные источники информации

1. Помощь пользователям системы Яндекс (www.yandex.ru).
2. Курсанов Д. Поиск. Отрывок из книги “Понятный Интернет” (<http://www.symbol.ru/dk/www.htm>).
3. Что такое поисковая машина, или Как работает поисковик (<http://site-stroy.com.ua/ru/article/search/28/85>).

Продолжение статьи читайте в следующем номере. Во второй части будет подробно рассмотрен язык запросов к поисковым системам.

VBA в приложении к Excel, Word и Power Point

О.А. ЖИТКОВА, Т.И. ПАНФИЛОВА,

Москва

Продолжение. Начало в № 1, 2/2006.

Занятие № 3. Практические работы с применением макросов

Практическая работа № 3–1 “Годовая температура”

Порядок работы:

1. Откройте Excel.
2. Переименуйте “Лист1” в лист “Температура”.
3. Составьте таблицу “Годовая температура”.

Месяц	Средняя температура
Январь	-5
Февраль	-6
Март	0
Апрель	2
Май	10
Июнь	15
Июль	18
Август	20
Сентябрь	18
Октябрь	6
Ноябрь	3
Декабрь	-9

4. В ячейку E3 поместите значение среднегодовой температуры, рассчитав ее по формуле СРЗНАЧ(B3:D3).

5. Запишите ваши действия в макрос, назвав его “СрГодТемп”.

6. Привяжите макрос к графическому объекту (например, “Кнопка”):

Отформатируйте кнопку и назначьте ей макрос “СрГодТемп”. 

7. Справа от показателя температуры нарисуйте графический объект (кнопку) и назначьте ему макрос, который будет очищать содержимое ячейки E3.

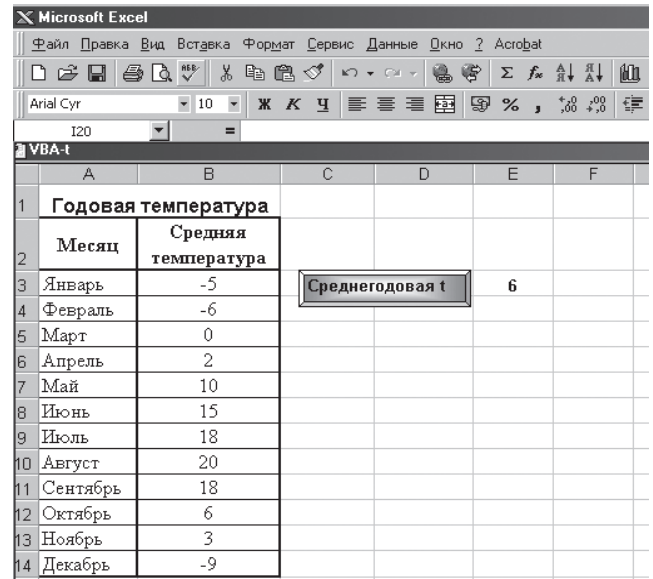
8. Проверьте работу кнопок “Среднегодовая t” и “Очистить”.

9. Самостоятельно выполните создание кнопок и макросов для подсчета и очистки:

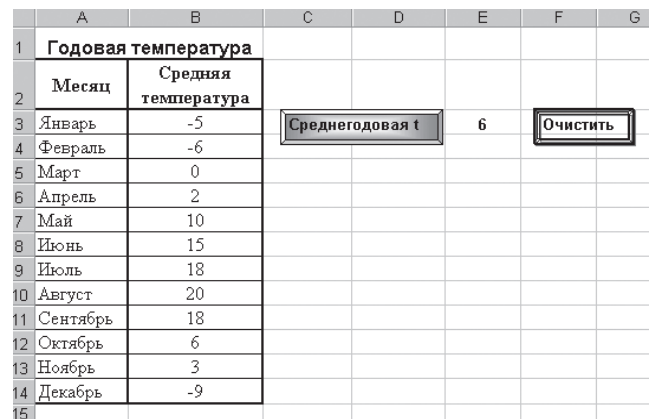
- Максимальной годовой температуры;
- Минимальной годовой температуры;
- Средней температуры по временам года (зимой, весной, летом, осенью);

• Создайте кнопку “Очистить все” для очистки всех расчетов.

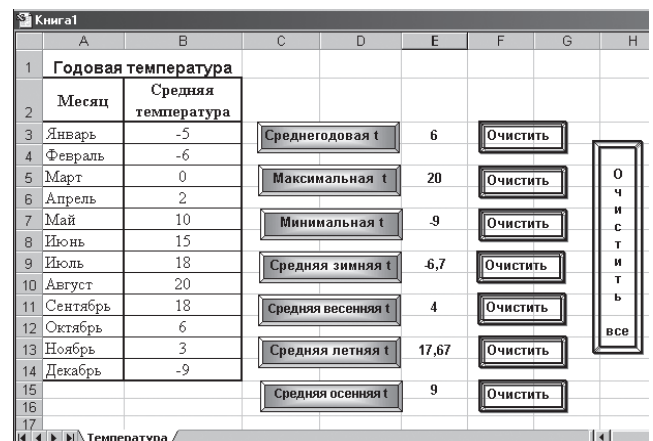
Ваша работа должна выглядеть так, как показано справа внизу.



Годовая температура					
Месяц	Средняя температура				
Январь	-5	Среднегодовая t	6		
Февраль	-6				
Март	0				
Апрель	2				
Май	10				
Июнь	15				
Июль	18				
Август	20				
Сентябрь	18				
Октябрь	6				
Ноябрь	3				
Декабрь	-9				



Годовая температура						
Месяц	Средняя температура					
Январь	-5	Среднегодовая t	6	Очистить		
Февраль	-6					
Март	0					
Апрель	2					
Май	10					
Июнь	15					
Июль	18					
Август	20					
Сентябрь	18					
Октябрь	6					
Ноябрь	3					
Декабрь	-9					



Годовая температура								
Месяц	Средняя температура							
Январь	-5	Среднегодовая t	6	Очистить			Очистить все	
Февраль	-6	Максимальная t	20	Очистить				
Март	0	Минимальная t	-9	Очистить				
Апрель	2	Средняя зимняя t	-6,7	Очистить				
Май	10	Средняя весенняя t	4	Очистить				
Июнь	15	Средняя летняя t	17,67	Очистить				
Июль	18	Средняя осенняя t	9	Очистить				
Август	20							
Сентябрь	18							
Октябрь	6							
Ноябрь	3							
Декабрь	-9							

Практическая работа № 3–2 “Протяженность границ РФ”

Порядок работы:

1. Откройте Excel.
2. Переименуйте “Лист2” в лист “Границы”.
3. Составьте таблицу “Протяженность государственной границы РФ, км”.

Участок границы	Сухопутная	Речная и озерная	Морская
С Норвегией	43	152,8	23,3
С Финляндией	1091,7	180,1	54
С Эстонией	89,5	235,3	142
С Латвией	137,2	133,3	
С Литвой	29,9	236,1	22,4
С Польшей	203,3	0,8	32,2
С Белоруссией	857,7	381,3	
С Украиной	1500,2	425,6	320
С Грузией	819,4	56,1	22,4
С Азербайджаном	272,4	55,2	22,4
С Казахстаном	5936,1	1576,7	85,8
С Монголией	2878,6	606,4	
С Китаем	650,3	3559	
С КНДР		17,3	22,1
С Японией			194,3
С США			49

4. Начиная с ячейки F3, составьте таблицу вопросов и ответов:

	Вопрос	Ответ
1	Суммарная протяженность морских границ	
2	Суммарная протяженность сухопутных границ	
3	Суммарная протяженность речных и озерных границ	
4	Максимальная длина границы	
5	Минимальная длина границы	
6	Самая протяженная граница (вид)	
7	Самая малая граница (вид)	

5. Справа от таблицы нарисуйте графические объекты, к которым привяжите макросы.

	A	B	C	D	E	F	G	H	I
1	«Протяженность государственной границы РФ, км»								
	Участок границы	Сухопутная	Речная и озерная	Морская					
2							ВОПРОС	ОТВЕТ	
3	С Норвегией	43	152,8	23,3	1	Суммарная протяженность морских границ	989,9		
4	С Финляндией	1092	180,1	54	2	Суммарная протяженность сухопутных границ	14509,3		
5	С Эстонией	89,5	235,3	142	3	Суммарная протяженность речных и озерных границ	7616		
6	С Латвией	137,2	133,3		4	Максимальная длина границы	5936,1		

6. Для расчетов используйте формулы:

в ячейке H3: =СУММ(D3:D18);

в ячейке H4: =СУММ(B3:B18);

в ячейке H5: =СУММ(C3:C18);

в ячейке H6: =МАКС(B3:D18);

в ячейке H7: =МИН(B3:D18);

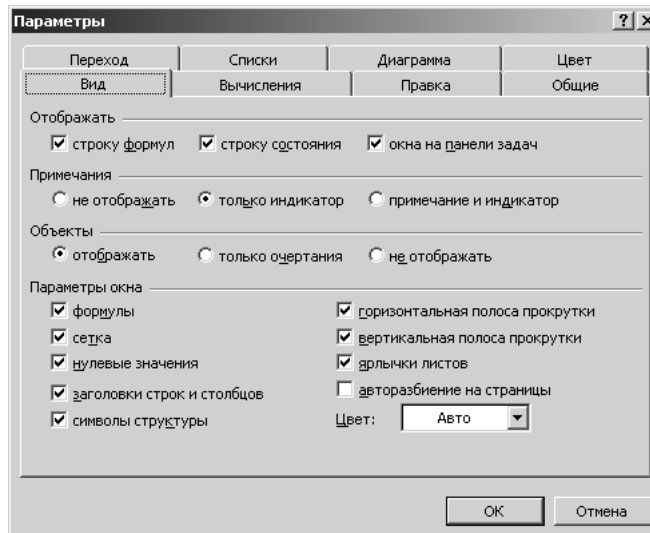
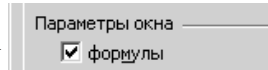
в ячейке H8: =ЕСЛИ(H6=МАКС(H6:H7);B2;ЕСЛИ(H6=МАКС(C3:C18);C2;D2));

в ячейке H9: =ЕСЛИ(H6=МИН(H6:H7);B2;ЕСЛИ(H6=МИН(C3:C18);C2;D2)).

После привязки макросов к графическим объектам и их выполнения получим результат:

VBA-t				ВОПРОС		ОТВЕТ
Участок границы	Сухопутная	Речная и озерная	Морская			
С Норвегией	43	152,8	23,3	1	Суммарная протяженность морских границ	989,9
С Финляндией	1092	180,2	54	2	Суммарная протяженность сухопутных границ	14509,3
С Эстонией	89,5	236,3	14,2	3	Суммарная протяженность речных и озерных границ	7616
С Латвией	137,2	133,3		4	Максимальная длина границы	5936,1
С Литвой	29,9	236,2	22,4	5	Минимальная длина границы	0,8
С Польшей	203,3	0,8	32,2	6	Самая протяженная граница (вид)	Сухопутная
С Белоруссией	857,7	381,3		7	Самая малая граница (вид)	Морская
С Украиной	1500	426,6	320			
С Грузией	819,4	56,2	22,4			
С Азербайджаном	272,4	55,2	22,4			
С Казахстаном	5936	1577	85,8			
С Монголией	2879	606,4				
С Китаем	650,3	3559				
С КНДР		17,3	22,1			
С Японией			194,3			
С США			49			

7. Выполните последовательно команды меню “Сервис | Параметры | Вкладка | Вид” и установите параметры окна для просмотра формул



8. Вы увидите все формулы, которые можно было ранее посмотреть только в строке формул.

ВОПРОС		ОТВЕТ
1	Суммарная протяженность морских границ	=СУММ(D3:D18)
2	Суммарная протяженность сухопутных границ	=СУММ(B3:B18)
3	Суммарная протяженность речных и озерных границ	=СУММ(C3:C18)
4	Максимальная длина границы	=МАКС(B3:D18)
5	Минимальная длина границы	=МИН(B3:D18)
6	Самая протяженная граница (вид)	=ЕСЛИ(Н6=МАКС(Н6:Н7);В2;ЕСЛИ(Н6=МАКС(С3:С18);С2;D2))
7	Самая малая граница (вид)	=ЕСЛИ(Н6=МИН(Н6:Н7);В2;ЕСЛИ(Н6=МИН(С3:С18);С2;D2))

- Убрав флажок, восстановите прежнее состояние листа.
- Создайте макрос, который очистит содержимое ячеек Н3–Н9, и привяжите его к графическому объекту.
- Проверьте работоспособность всех макросов, привязанных к графическим объектам.

Практическая работа № 3–3 “Воспроизводство населения России (%)”

Порядок работы:

1. Откройте Excel.
2. Переименуйте “Лист3” в лист “Рождаемость”.
3. Составьте таблицу “Воспроизводство населения России (%)”.

Год	Рождаемость	Смертность	Естественный прирост
1926	44,7	21,3	23,4
1940	33	20,6	12,4
1960	23,2	7,4	15,8
1980	15,9	10,6	5,6
1989	14,6	10,6	4,0
1991	12,1	11,4	0,7
1993	9,4	14,5	-5,1
1995	9,3	15,0	-5,7
1997	8,6	13,8	-5,2

4. Под таблицей выпишите вопросы и предусмотрите место для вывода ответов. Ответы должны выводиться при нажатии на графические объекты, к которым привязаны макросы.

Вопрос	Ответ
Средняя довоенная рождаемость (1926–1940 гг.)	
Средняя послевоенная рождаемость (1960–1997 гг.)	
Средняя довоенная смертность (1926–1940 гг.)	
Средняя послевоенная смертность (1960–1997 гг.)	
Средний довоенный естественный прирост (1926–1940 гг.)	
Средний послевоенный естественный прирост (1960–1997 гг.)	
Минимальная рождаемость	
Минимальная смертность	
Минимальный естественный прирост	
Максимальная рождаемость	
Максимальная смертность	
Максимальный естественный прирост	

5. Ваш лист должен выглядеть так, как показано вверху справа.

6. После записи макросов ВОПРОС и ОТВЕТ проверьте текст (“Сервис | Макрос | Макросы | Изменить”):

```
Sub ВОПРОС ()
' ВОПРОС Макрос
' Макрос записан 17.12.2005
'
Range("F15").Select
ActiveCell.FormulaR1C1 = "=AVERAGE(R[-12]C[-4]:R[-11]C[-4])"
Range("F16").Select
ActiveCell.FormulaR1C1 = "=AVERAGE(R[-11]C[-4]:R[-5]C[-4])"
Range("F17").Select
ActiveCell.FormulaR1C1 = "=AVERAGE(R[-14]C[-3]:R[-13]C[-3])"
```

```
Range("F18").Select
ActiveCell.FormulaR1C1 = "=AVERAGE(R[-13]C[-3]:R[-7]C[-3])"
Range("F19").Select
ActiveCell.FormulaR1C1 = "=AVERAGE(R[-16]C[-2]:R[-15]C[-2])"
Range("F20").Select
ActiveCell.FormulaR1C1 = "=AVERAGE(R[-15]C[-2]:R[-9]C[-2])"
Range("F21").Select
ActiveCell.FormulaR1C1 = "=MIN(R[-18]C[-4]:R[-10]C[-4])"
Range("F22").Select
ActiveCell.FormulaR1C1 = "=MIN(R[-19]C[-3]:R[-11]C[-3])"
Range("F23").Select
ActiveCell.FormulaR1C1 = "=MIN(R[-20]C[-2]:R[-12]C[-2])"
Range("F24").Select
ActiveCell.FormulaR1C1 = "=MAX(R[-21]C[-4]:R[-13]C[-4])"
Range("F25").Select
ActiveCell.FormulaR1C1 = "=MAX(R[-22]C[-3]:R[-14]C[-3])"
Range("F26").Select
ActiveCell.FormulaR1C1 = "=MAX(R[-23]C[-2]:R[-15]C[-2])"
End Sub
```

```
Sub ОЧ ()
' ОЧ Макрос
' Макрос записан 17.12.2005 (*)
' Range("F15:F26").Select
Selection.ClearContents
End Sub
```

7. Создайте макросы изменения цвета таблиц на зеленый и изменения шрифта на полужирный или курсив.

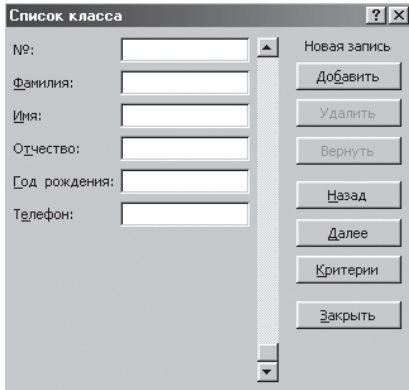
8. Привяжите эти макросы к графическим объектам.

Практическая работа № 3–4 “Количество осадков”

Порядок работы:

1. Откройте Excel.
2. Переименуйте “Лист 4” в лист “Осадки”.
3. Составьте таблицу “Количество осадков (мм)”.

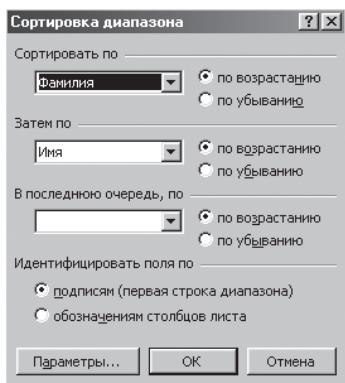
- Выполнить команду меню “Данные | Форма”. Появится форма, которую надо заполнить.



5. Создайте макрос вызова автоформы и привяжите его к графическому объекту.
6. Заполните таблицу:

№	Фамилия	Имя	Отчество	Год рождения	Телефон
1	Тернова	Галина	Викторовна	1988	654-5444
2	Петров	Юрий	Олегович	1990	234-4455
3	Яковлев	Сергей	Анатольевич	1991	456-6789
4	Иванова	Анна	Юрьевна	1996	123-4433
5	Аксаков	Алексей	Иванович	1990	122-3344
6	Горбачева	Ольга	Ивановна	1989	122-2345
7	Ильин	Георгий	Александрович	1988	345-6655
8	Байбурина	Наталья	Леонидовна	1987	133-5555
9	Леонидов	Иван	Игоревич	1980	122-6666
10	Луговой	Петр	Николаевич	1987	345-5566
11	Глебов	Петр	Алексеевич	1989	987-7777
12	Лесной	Станислав	Иванович	1990	654-6666
13	Аксенова	Ольга	Николаевна	1989	147-4017
14	Мелова	Татьяна	Глебовна	1985	234-4478
15	Зайцева	Ирина	Витальевна	1989	765-45-34

7. Создайте макрос сортировки таблицы по фамилии и имени (с помощью выделения диапазона ячеек B1–F16), привяжите его к геометрическому объекту.



8. Создайте макрос сортировки по годам и фамилиям, привяжите его к геометрическому объекту.
9. Создайте макрос (с привязкой к графическому объекту) изменения цвета и шрифта в диапазоне:

- A1–F1 — полужирный шрифт, кегль 16, зеленый фон, буквы белые;
 - A2–A16 — розовый фон;
 - B2–B16 — голубой фон;
 - C2–C16 — серый фон;
 - D2–D16 — розовый фон;
 - E2–E16 — голубой фон;
 - F2–F16 — серый фон.
10. Создайте макрос (с привязкой к графическому объекту) возврата в исходное состояние.
 11. Проверьте работу макросов.

**Практическая работа № 3–6
“Преступления в сфере высоких технологий”**

Порядок работы:

1. Откройте Excel.
2. Переименуйте “Лист 6” в лист “Преступления”.
3. Составьте таблицу “Преступления в сфере высоких технологий”.

Виды преступных деяний	Годы			
	1998	1999	2000	2001
Проникновение в чужие компьютерные сети	22 145	21 568	18 900	17 895
Распространение вредоносных программ	45 475	46 879	51 057	60 458
Нарушение работы ЭВМ и баз данных	45 879	45 487	44 568	45 757
ИТОГО по преступлениям в сфере высоких технологий				

4. Определите для всей таблицы в целом:
 - Суммарное количество преступлений за 1998, 1999, 2000 и 2001 годы (заполните строку “Итого по преступлениям в сфере высоких технологий”);
 - Максимальное количество преступлений за 4 года;
 - Минимальное количество преступлений за 4 года;
 - Суммарное количество проникновений в чужие компьютерные сети за все 4 года;
 - Суммарное количество распространения вредоносных программ за все 4 года;
 - Суммарное количество нарушений работы ЭВМ и баз данных за все 4 года;
 - Год с наибольшим количеством преступлений по итоговому результату;
 - Год с наименьшим количеством преступлений по итоговому результату;
 - Наиболее распространенный вид преступных деяний.

Результаты представьте в виде дополнительной таблицы.

Продолжение в следующем номере.

Задача конкурса "КИО-2005" "Автомат для голосования"

С.Н. ПОЗДНЯКОВ, Д.О. СТЕПУЛЕНКО,
Санкт-Петербург

Конкурс "КИО-2005" (Конструируй, Исследуй, Оптимизируй) прошел в феврале 2005 года. Информацию о предыдущем конкурсе "КИО-2004" и разбор задач можно найти в "Информатике" № 5/2005.

Задачи конкурса имеют исследовательский характер. В привычном смысле слова каждая задача имеет множество решений, но нужно постараться найти как можно лучшее из них по указанному в задаче признаку. Таким образом, каждый участник конкурса мог предложить свое решение. Победители определялись по тому, насколько найденные ими решения оказались лучше решений, присланных другими участниками.

Подобные задачи обладают следующими свойствами.

1. Для задач этого класса не известны эффективные непереборные алгоритмы.

2. Для этих задач легко ищется неоптимальное (эвристическое) решение, но для нахождения оптимального решения нужны идеи, которые сродни интеллектуальным действиям людей, совершающих математические открытия.

3. Для данных задач можно сконструировать программный инструмент, который берет на себя выполнение технических операций, что позволяет ученику проводить эксперименты в процессе решения задачи. Кроме того, инструмент выполняет некоторые вспомогательные математические операции. Инструменты, осуществляющие эти операции, должны быть "прозрачными" для ученика и не оставлять впечатления "черного ящика".

4. Эти задачи допускают обобщение. Решения некоторых из них в общем виде могут быть не известны до сих пор. Предоставляемый инструмент должен давать возможность не только решения данных задач, но и конструирования новых, близких в идейном отношении.

Рассмотрим одну из задач конкурса "Автомат для голосования".

"Имеется неограниченное число логических элементов для голосования с тремя входами и одним выходом (если на двух или трех входах такого элемента 1, то на выходе 1, иначе на выходе 0). Если один из входов логического элемента остается свободным, то на этом входе всегда ноль. Из них надо собрать устройство для голосования жюри из 7 человек, используя наименьшее число элементов (на наборах вход-

ных сигналов, содержащих более трех 1, устройство должно выдавать 1, иначе 0). Правильно работающие устройства сравниваются по числу элементов.

Устройства, которые работают неправильно, сравниваются по доле наборов, на которых они дают правильный результат. При равной доле лучше тот, у которого меньше элементов".

Предложенные элементы реализуют функцию доминирования. Эту функцию можно записать, используя только две операции: конъюнкцию и дизъюнкцию:

$$f(x; y; z) = xy \vee xz \vee yz.$$

Действительно, если две из трех переменных равны 1, то и результат будет равен 1.

Попробуем построить функцию доминирования для 5 переменных (автомат для 5 голосующих):

$$f(x; y; z; u; v) = xyz \vee xyu \vee xuv \vee xzu \vee xzv \vee \vee xiv \vee yzu \vee yzv \vee yiv \vee ziv.$$

Нетрудно видеть, что в правой части формулы фактически перечислены все варианты голосования, оканчивающиеся принятием положительного решения.

Один из кажущихся перспективным путей решения — сделать из предлагаемых элементов автоматы, реализующие конъюнкцию и дизъюнкцию, а затем из них собрать автомат для любого количества голосующих.

Упражнение. Попробуйте сделать это или доказать невозможность такого решения.

Указание. Попробуем собрать схему с двумя входами из одного элемента. Вариантов не много: в любом из них надо подсоединить один из двух входов (дизъюнкция и конъюнкция — функции двух переменных) к двум входам элемента (рис. 1), но в этом случае на выходе всегда будет то же значение, которое принимает переменная, значение которой подается на два входа. Таким образом, мы получаем одну из двух функций:

$$f(x; y) = x \text{ или } f(x; y) = y,$$

которые не приводят к интересным результатам.

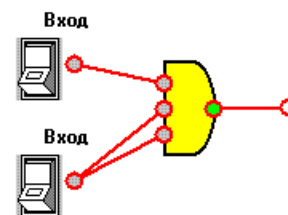


Рис. 1

Попробуем построить искомый автомат, соединяя два базовых элемента в различных допустимых вариантах. Их также не так много (один из них показан на *рис. 2*).

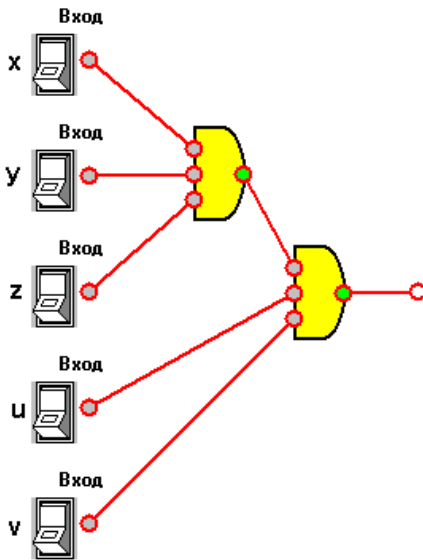


Рис. 2

Если на выходе первого (стоящего левее) элемента значение 1, то две из трех переменных x , y , z также принимают значения 1. Тогда для положительного голосования достаточно, чтобы хотя бы одна из переменных x и y принимала значение 1. Неправильный результат схема даст, только если “за” голосуют xuz , на остальных 9 тройках схема дает правильный результат.

Это подсказывает такое решение задачи для 5 голосующих (*рис. 3*).

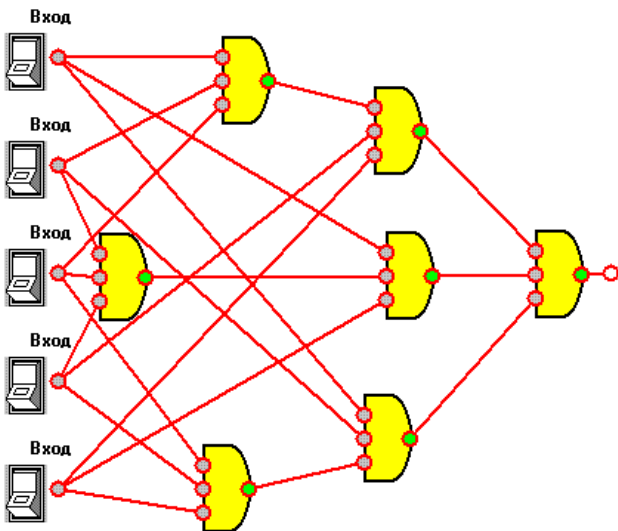


Рис. 3

Нетрудно видеть, что схема получена соединением трех частей, построенных выше, посредством еще одного элемента доминирования.

Первая из частей дает неправильный результат только на тройке xuz , вторая — на uzi , третья — на ziv . Однако, так как в это же время две другие части дают на этих элементах значение 1, то и на выходе замыкающего элемента будет значение 1. Остальные 7 наборов, определяющих большинство, дают значение 1 на выходе каждой из частей и, следовательно, дадут значение 1 и на выходе замыкающего элемента доминирования. Теперь надо проверить, не дает ли схема положительного результата голосования на наборах, у которых не более двух переменных принимают значения 1. В силу симметричности схемы достаточно проверить, что схема дает нули, если голосуют “за” только xu или xz или xi или xv . Это нетрудно сделать вручную.

Заметьте, что количество элементов в схеме можно уменьшить.

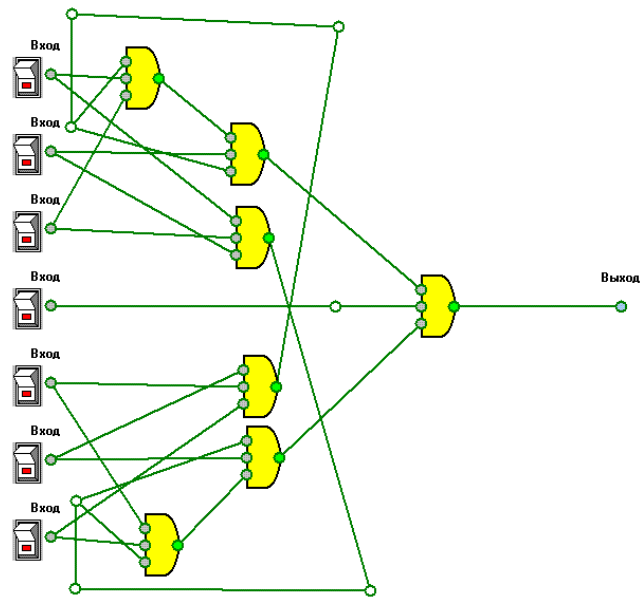


Рис. 4

Лучшее решение задачи для 7 членов жюри приведено на *рис. 4*. Это решение было найдено жюри конкурса (автор — Д.О. Степуленок). Однако нет уверенности в том, что и это решение нельзя улучшить. Возможно, это удастся кому-то из читателей?

Подробную информацию о конкурсах КИО можно найти на сайте <http://www.ipospb.ru/kio>.

Больше, чем долина

Л.Н. КАРТВЕЛИШВИЛИ,
Москва

Трудно “найти человека, который не слышал бы словосочетания *Кремниевая долина* — неформального названия узкой полоски земли, которую можно проехать вдоль из конца в конец всего за полчаса. Гораздо меньше известно о происхождении этого названия” [1].

“В феврале 1956 года один из создателей транзистора Уильям Брэдфорд Шокли основал в Пало-Альто научно-производственную компанию Shockley Semiconductor Laboratories, Inc. (SSL). Считается, что с этого времени началась история знаменитого научно-промышленного ареала, названного позже Кремниевой долиной. Но это не совсем верно...” [2].

В конце октября 1769 года экспедиция испанца Гаспара де Портолы, исследовавшая западное побережье Северной Америки в районе Калифорнии, высадилась на берег и расположилась лагерем в тени гигантской секвойи, которую участники экспедиции назвали *El Palo Alto* (исп. высокое дерево). По приказу капитана один из его помощников, Хосе Франциско Ортега, во главе небольшого отряда отправился на поиски удобной стоянки для корабля. Двигаясь к северу, он поднялся на вершину невысокой горы и увидел перед собой прекрасную бухту, отделенную от океана узким проходом. Оказалось, что земля, на которую высадились испанцы, представляет собой полуостров. В честь святого, покровителя Ортеги, эти полуостров, залив и позже город назвали именем св. Франциска (Сан-Франциско), а проход нарекли *Золотыми воротами*. Причем последнее случилось много лет спустя, когда Калифорния из мексиканской провинции превратилась в американский штат. Одним из его первых губернаторов стал в 1861 году юрист и предприниматель Леланд Станфорд. Через 15 лет он решил сменить место и образ жизни, купил большой участок земли в той местности, где когда-то располагался лагерь де Портолы, построил дом и занялся коннозаводством. Спустя еще несколько лет случилось несчастье: от брюшного тифа умер единственный сын Леланда Станфорда. В память о нем Леланд решил основать на землях своего ранчо университет, который “готовил бы студентов к полезной для общества жизни и достижению личного успеха, способ-

ствовал росту общественного благосостояния во имя гуманности и цивилизации, развивал уважение к дарованным законом свободам, прививал любовь и благоговение перед великими принципами правления, основанного на неотъемлемом праве человека на жизнь, свободу и поиски счастья”. Двери университета распахнулись в Пало-Альто 1 октября 1891 года.

Этот городок, расположенный примерно в 45 милях к юго-востоку от Сан-Франциско, — один из двух десятков населенных пунктов округа Санта-Клара, находящегося в узкой (около 25 миль в длину и 10 миль в ширину) долине того же названия. С запада горная гряда Санта-Крус защищает ее от холодного воздуха Тихого океана, с востока горы Сьерра-Невады преграждают путь горячему дыханию пустыни. Умеренно жаркий климат и свыше трехсот солнечных дней в году сделали долину превосходным местом для выращивания слив, абрикосов, миндаля, вишни и ряда других культур. Однако последние 40–50 лет округ Санта-Клара поставляет на американский и мировой рынки главным образом иные товары. Это и персональные компьютеры, и программное обеспечение, и интегральные схемы, и спутниковое оборудование, и биотехнологии, и многое другое. Здесь располагается множество компаний, связанных преимущественно с электронной и компьютерной индустрией. Данное обстоятельство дало основание калифорнийскому журналисту и издателю газеты *Microelectronics News* Дональду К. Хефлеру “переименовать” в 1971 году долину Санта-Клара в *Кремниевую долину* (Silicon Valley), ведь кремний — основной материал полупроводникового производства [2–7].

“Кремниевая долина — это больше, чем место. Это феномен, это стиль жизни, культура, конгломерат идей, луч надежды, набор возможностей, микрокосм потенциального будущего”, — вот слова одного из родоначальников микроэлектронной индустрии Роберта Нойса [2]. Чем же эта узкая полоска земли привлекала (и привлекает) ученых, инженеров, предпринимателей? Кто эти люди, ставшие в одночасье знаменитыми? Почему на площади, вдвое меньшей, чем княжество Монако, насчитывается



Мост “Золотые ворота”, Сан-Франциско, Калифорния, США

несколько десятков тысяч миллионеров?

Социологи, историки, специалисты по промышленной динамике расходятся во мнениях. Например, социолог Юджин Н. Ландрам отмечает: «Инноваторы Кремниевой долины отличаются уникальными качествами, которые редки для других регионов страны. Они пренебрегают традициями, самосохранением, идут наперекор запретам властей, принимают неординарные рискованные решения, интуитивно чувствуя все веяния... Их успех предначертан несокрушимым оптимизмом и верой в собственное представление о реальности... Их поведение в бизнесе отличается от общепринятых норм и вызвано подсознательным пониманием необходимости действовать вопреки сложившемуся стилю».

Возможно, Ландрам прав. Но, несомненно, следует добавить сюда еще такие факторы, как прекрасный климат, огромные вложения государства в развитие оборонной промышленности Калифорнии, благоприятную ситуацию на рынке труда, высокий средний уровень заработной платы, удобные транспортные артерии, наличие местных «поставщиков» высококвалифицированных кадров с университетскими дипломами. Большое значение имеет также тесная связь научного сообщества штата с его индустрией, и важнейшую роль в свое время тут сыграл Фредерик Эммонс Термен — профессор, а затем декан электротехнического факультета и вице-президент Станфордского университета. Как писал американский историк, «Кремниевая долина — это лишь удлиненная тень человека по фамилии Термен». Сам Термен выдающимся изобретателем не был, но он приложил значительные усилия, чтобы превратить Станфордский университет в одну из самых значительных научных и инженерных школ. В знак признания заслуг Термена его именем назван инженерный корпус в Станфорде.

Детские и юношеские годы Термен провел в Пало-Альто, а молодость — в стенах Станфорда. С 1927 года Термен начинает преподавать в этом учебном заведении. Он также возглавляет лабораторию, выполнявшую заказы различных компаний, пишет книги. Контакты с промышленниками убеждают его в необходимости наладить взаимодействие научного потенциала Станфорда с индустрией. Реализация данной идеи стала «стержнем» деятельности Термена в последующие годы. В период Великой депрессии он настойчиво убеждал выпускников университета не обивать пороги известных компаний в поисках работы, а организовывать собственное дело в калифорнийском регионе. И по крайней мере в одном случае агитация Термена блестяще удалась: два его бывших студента — Уильям Хьюлетт и Дэвид Паккард — создали компанию (Hewlett-Packard, или HP), являющуюся сегодня



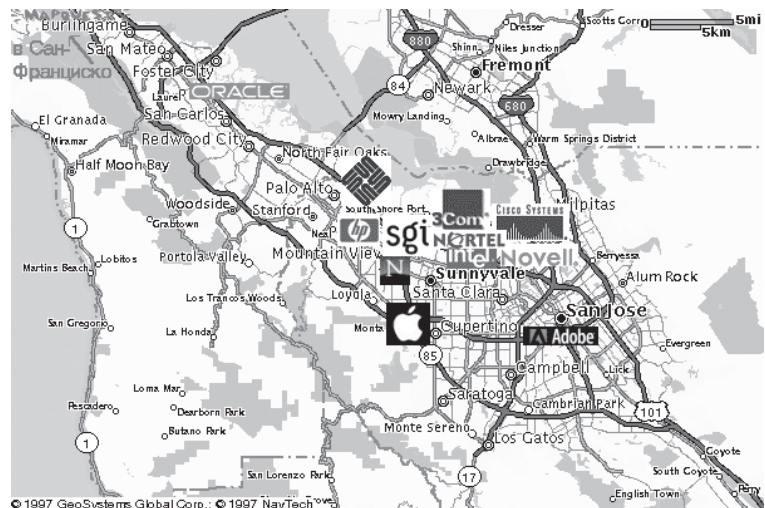
Партнеры: Дэвид Паккард (слева) и Уильям Хьюлетт. 1960 год

лаборатории такие гиганты, как General Electric, IBM, Xerox, Eastern Kodak, Lockheed Martin и другие. Позже там сформировались, например, такие широко известные ныне компании, как Intel, Seagate, Apple Computer, Oracle, Sun Microsystems, Silicon Graphics, Netscape Communications, Cisco Systems, Yahoo!, Google.

Сегодня в Кремниевой долине действует более четырех тысяч компаний [1]. Их совокупный доход превышает четверть триллиона долларов, а число занятых в них специалистов измеряется сотнями тысяч.

Литература

1. Черняк Л. Краткий курс истории Кремниевой долины // Computerworld № 36/2003.
2. Полунов Ю.Л. От абака до компьютера: судьбы людей и машин. Книга для чтения по истории вычислительной техники в двух томах. Т. II. М.: Издательско-торговый дом «Русская Редакция», 2004.
3. Знакомьтесь: компьютер: Пер. с англ. М.: Мир, 1989.
4. Толковый словарь по вычислительной технике (Microsoft Corporation): Пер. с англ. М.: Издательский отдел «Русская редакция» ТОО «Channel Trading Ltd.», 1995.
5. Частиков А.П. Архитекторы компьютерного мира. СПб.: БХВ-Петербург, 2002.
6. Частиков А.П. От калькулятора до суперЭВМ // Новое в жизни, науке, технике. Сер. «Вычислительная техника и ее применение» № 1/1988.
7. Путь к транзистору // Информатика № 6/2000.



«Кремниевая долина», Калифорния, США

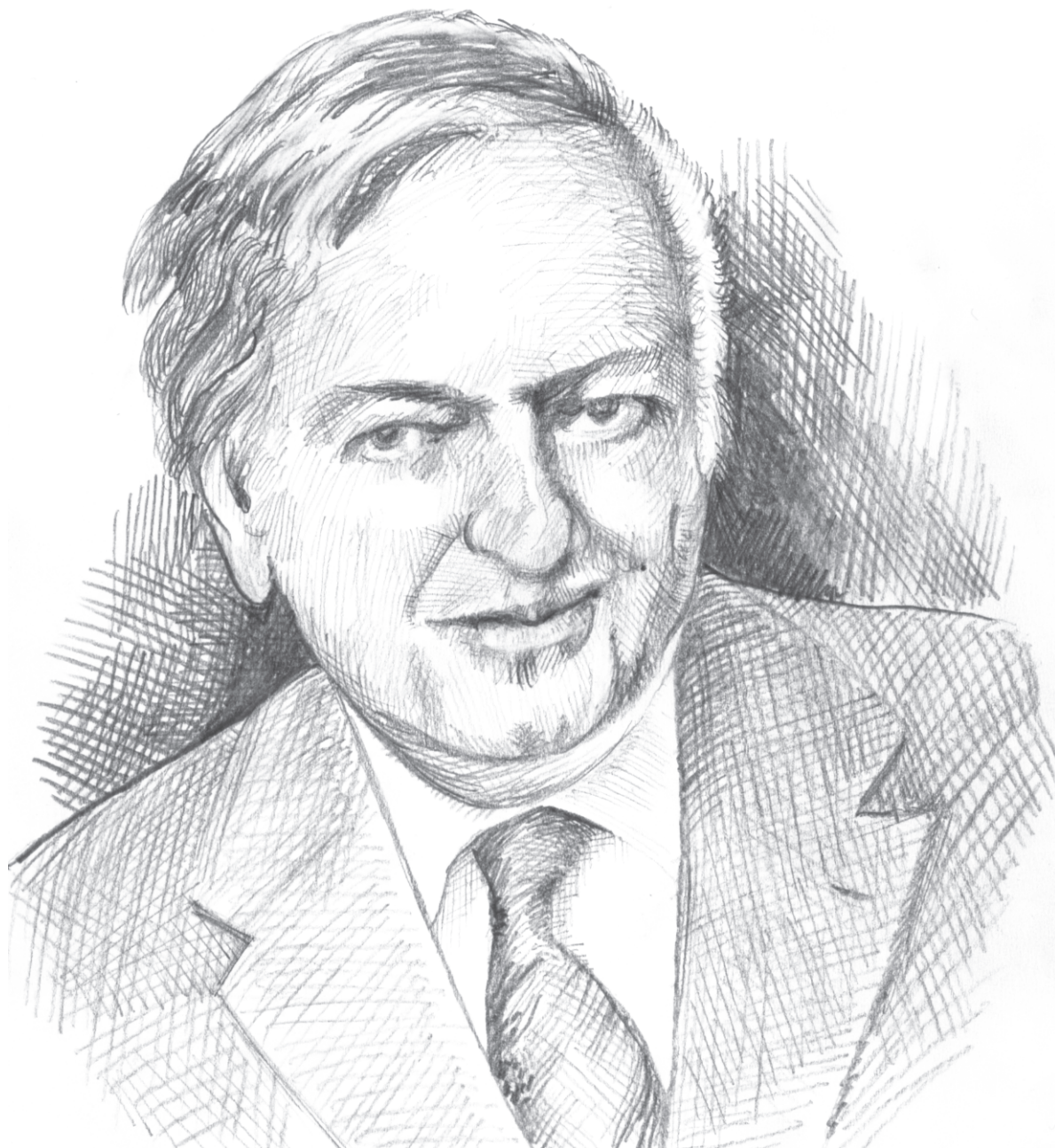


ДЖОН (ЯНОШ) КЕМЕНИ

**John (Janos) George Kemeny
(31 мая 1926 – 26 декабря 1992)**

Математик, профессор Дартмутского колледжа (США). Вместе с Томасом Курцем разработал язык программирования BASIC и сетевую систему пользования несколькими компьютерами одновременно ("time sharing").

Вместе с родителями эмигрировал в США из Венгрии в 1940 году. Окончил Принстонский университет, где изучал математику и философию. В 1949 году защитил диссертацию, а в 1953 году был приглашен в Дартмут. Будучи деканом Математического факультета Дартмутского колледжа с 1955 по 1967 год и даже находясь на посту президента колледжа (1970–1981), не оставлял преподавательской деятельности. Явился одним из пионеров преподавания основ программирования: считал, что этот предмет должен быть доступен всем студентам, вне зависимости от их специализации.



ТОМАС КУРЦ

Thomas Eugene Kurtz

Родился 28 февраля 1928

Математик, профессор Дартмутского колледжа (США). Вместе с Джоном Кемени разработал язык программирования BASIC и сетевую систему пользования несколькими компьютерами одновременно ("time sharing").

Получил научную степень в Принстонском университете в 1956 году и сразу был приглашен на Математический факультет в Дартмутский колледж. Наряду с Джоном Кемени считал, что студенты должны осваивать машину, создавая работающие программы, а не просто слушая лекции. В результате такого подхода появился многозадачный язык символьных конструкций для начинающих — BASIC (The Beginner's All-Purpose Symbolic Instruction Code). Кемени и Курц не стали патентовать свое изобретение, что способствовало его развитию и увеличению числа версий.

“Рождественским чтениям” — 10 лет*

В Перми прошла юбилейная конференция учителей информатики

В начале января, сразу после рождественских праздников, в Пермском государственном университете прошла традиционная конференция учителей информатики, которой в этом году исполнилось уже 10 лет. Все эти годы оргкомитет конференции возглавляет профессор С.В. Русаков. Во многом благодаря его убежденности в необходимости поддержания традиции и его организаторской энергии конференция существует все эти годы.

“Рождественские чтения” зародились на волне развития феномена пермской версии школьного курса информатики, которая имеет свою самостоятельную и содержательную историю.

В 1993 г. согласно новому Базисному учебному плану информатика была переведена из старших классов в основную школу. Не многие знают, что в Федеральном БУПе в то время информатика была поставлена в 7-й класс в объеме 68 часов. Пермский Департамент образования принял решение преподавать информатику в 8–9-х классах в том же объеме часов. Правда, вместо двух уроков в неделю получалось по одному уроку. Но такой план позволял организовать непрерывное обучение предмету в случае, если в 10–11-х классах изучение информатики будет продолжено за счет школьного резерва учебного времени. Именно в те годы начинала формироваться идея профильной подготовки учащихся в старших классах.

Практически во всех пермских школах с 1 сентября 1993 г. информатику начали преподавать с 8-го класса. В то время не существовало подходящей учебной литературы по информатике для основной школы. Кроме того, назрела необходимость пересмотра содержания школьного курса информатики. В 1992 г. в Законе РФ “Об образовании” была провозглашена концепция образовательных стандартов. Началась разработка ГОС по всем школьным предметам. Для информатики, как, может быть, ни для какого другого предмета, эта задача была чрезвычайно актуальной.

Сработал известный принцип: “Спасение утопающих — дело рук самих утопающих”. Пермским школам срочно требовалось учебно-методическое обеспечение для преподавания информатики в 8–9-х классах, поэтому разрабатывать его нужно было самим. Для решения этой проблемы областной Департамент образования сформировал рабочую группу. В нее вошли сотрудники госуниверситета и педагогического института: Л.А. Залогова, С.В. Русаков, О.Л. Русакова, И.Г. Семакин, Е.К. Хеннер, Л.В. Шестакова, Т.Ю. Шеина. Ими были подготовлены методические указания для учителей и рабочая тетрадь для учащихся.

* При подготовке информационного сообщения использованы материалы выступления И.Г. Семакина на открытии конференции.

Конференцию открывает С.В. Русаков



Нельзя сказать, что переход к преподаванию нового курса информатики прошел в школах безболезненно. Многим учителям приходилось отказываться от принципа “преподаю, что хочу” в пользу принципа “преподаю, что надо”. Еще не была отработана методика преподавания новых разделов курса, прежде всего — информационных технологий, некоторых вопросов теоретической информатики. Очень разношерстным был компьютерный парк, программное обеспечение. Фактически в рамках Пермской области шел большой эксперимент по обновлению содержания школьной информатики, по отработке методики ее преподавания. Каждый учитель здесь играл роль исследователя-методиста. Поэтому можно говорить о том, что “строителями” пермской версии базового курса информатики были не только упомянутые выше специалисты, но и большое число пермских учителей.

После трех лет обширного педагогического эксперимента возникла необходимость подведения итогов, обмена опытом, публикации материалов о наиболее интересных педагогических достижениях учителей. Вторым учебников была необходима “обратная связь”, которая позволила бы совершенствовать содержание курса. С этой целью в 1997 г. была организована первая областная конференция “Рождественские чтения”. С тех пор конференция успешно выполняет свою главную функцию — систематизацию деятельности сообщества пермских педагогов-информатиков, специалистов, работающих в области информатизации образования.

За прошедшее время произошло много событий вокруг школьной информатики и информатизации. Можно говорить о том, что сформировалась пермская научно-методическая школа в области информатики. Пермская версия базового курса из региональной превратилась в федеральную. Возможно, в самом ближайшем будущем статус федеральной получит и конференция “Рождественские чтения”.

С материалами конференции можно познакомиться на сайте <http://uniserv.math.psu.ru/conf>.

Системы счисления и их применения*

С.Б. ГАШКОВ,

Москва

Продолжение. Начало в № 2/2006.

§ 4. Краткая история двоичной системы

Некоторые идеи, лежащие в основе двоичной системы, по существу были известны в Древнем Китае. Об этом свидетельствует классическая книга “И цзин” (“Книга перемен”), о которой речь пойдет позже.

Идея двоичной системы была известна и древним индусам.

В Европе двоичная система, видимо, появилась уже в новое время. Об этом свидетельствует система объемных мер, применяемая английскими виноторговцами: два джилла = полуштоф, два полуштофа = пинта, две пинты = кварта, две кварталы = потл, два потла = галлон, два галлона = пек, два пека = полубушель, два полубушеля = бушель, два бушеля = килдеркин, два килдеркина = баррель, два барреля = хогзед, два хогзед = пайп, два пайпа = тан.

Читатели исторических романов, видимо, знакомы с пинтами и квартами. Частично эта система дожила и до нашего времени (нефть и бензин до сих пор меряют галлонами и баррелями).

И в английских мерах веса можно увидеть двоичный принцип.

Так, фунт (обычный, не тройский) содержит 16 унций, а унция — 16 дрэмов. Тройский фунт содержит 12 тройских унций. В английских аптекарских мерах веса, однако, унция содержит восемь дрэмов.

Пропагандистом двоичной системы был знаменитый Г.В. Лейбниц (получивший, кстати, от Петра I звание тайного советника).

Он отмечал особую простоту алгоритмов арифметических действий в двоичной арифметике в сравнении с другими системами и придавал ей определенный философский смысл. Говорят, что по его предложению была выбита медаль с надписью: “Для того чтобы вывести из ничтожества все, достаточно единицы”. Известный современный математик Т.Данциг о нынешнем положении дел сказал: “Увы! То, что некогда возвышалось как монумент монотеизму, очутилось в чреве компьютера”. Причина такой метаморфозы не только уникальная простота таблицы умножения в двоичной системе, но и особенности физических принципов, на основе которых работает элементная база современных ЭВМ (впрочем, за последние 40 лет она неоднократно менялась, но двоичная система и булева алгебра по-прежнему вне конкуренции).

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

Двумя чертами слева выделены тексты задач для самостоятельного решения.

§ 5. Почему двоичная система удобна?

Главное достоинство двоичной системы — простота алгоритмов сложения, вычитания, умножения и деления. Таблица умножения в ней совсем не требует ничего запоминать: ведь любое число, умноженное на ноль, равно нулю, а умноженное на единицу равно самому себе. И при этом никаких переносов в следующие разряды, а они есть даже в троичной системе. Таблица деления сводится к двум равенствам $0/1 = 0$, $1/1 = 1$, благодаря чему деление столбиком многозначных двоичных чисел делается гораздо проще, чем в десятичной системе и, по существу, сводится к многократному вычитанию.

Таблица сложения, как ни странно, чуть сложнее, потому что $1 + 1 = 10$ и возникает перенос в следующий разряд. В общем виде операцию сложения однобитовых чисел можно записать в виде $x + y = 2w + v$, где w, v — биты результата. Внимательно посмотрев на таблицу сложения, можно заметить, что бит переноса w — это просто произведение xu , потому что он равен единице лишь когда x и y равны единице. А вот бит v равен $x + y$, за исключением случая $x = y = 1$, когда он равен не 2, а 0. Операцию, с помощью которой по битам x, y вычисляют бит v , называют по-разному. Мы будем использовать для нее название “сложение по модулю 2” и символ \oplus . Таким образом, сложение битов выполняется фактически не одной, а двумя операциями.

Если отвлечься от технических деталей, то именно с помощью этих операций и выполняются все операции в компьютере.

Для выполнения сложения однобитовых чисел делают обычно даже специальный логический элемент с двумя входами x, y и двумя выходами w, v , как бы составленный из элемента умножения (его часто называют конъюнкцией, чтобы не путать с умножением многозначных чисел) и элемента сложения по модулю 2. Этот элемент часто называют полусумматором.

§ 6. Ханойская башня, код Грея и двоичный n -мерный куб

Далее мы рассмотрим несколько интересных задач, в решении которых помогает знание двоичной системы. Начнем мы с задач, в которых используется только одна, самая простая из лежащих в ее основе идей — идея чисто комбинаторная и почти не связанная с арифметикой.

Первая из них — это “Ханойская башня”. Головоломку под таким названием придумал французский математик Эдуард Люка в XIX веке.

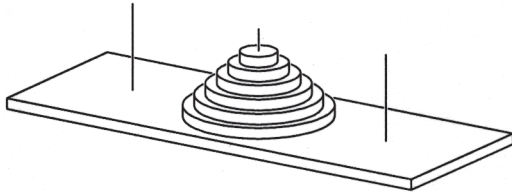


Рис. 2

На столбик нанизаны в порядке убывания размеров n круглых дисков с дырками в центре в виде детской игрушечной пирамидки. Требуется перенести эту пирамидку на другой столбик, пользуясь третьим вспомогательным столбиком (рис. 2). За один ход разрешается переносить со столбика на столбик один диск, но класть больший диск на меньший нельзя. Спрашивается, за какое наименьшее количество ходов это можно сделать? Ответом в этой задаче служит уже известное нам “индийское число” $2^n - 1$. Люка в своей книге приводит якобы известную легенду о том, что монахи в одном из монастырей Ханоя занимаются перенесением на другой столбик пирамидки, состоящей из 64 дисков. Когда они закончат работу, кончится жизнь Брахмы⁵. Видно, ждать придется долго.

Решение этой головоломки сильно облегчается, если знать, что такое код Грея. Кодом Грея порядка n называется любая циклическая последовательность всех наборов из нулей и единиц длины n , в которой два соседних набора отличаются ровно в одной компоненте. Примером кода Грея порядка 3 является последовательность трехразрядных наборов 000, 001, 011, 010, 110, 111, 101, 100.

9. Докажите, что длина кода Грея порядка n равна 2^n .

Если занумеровать компоненты каждого набора справа налево (при этом последняя, т.е. самая правая, компонента получит номер 1) и начинать код Грея с нулевого набора, то его можно записать короче, если вместо очередного набора писать только номер компоненты, в которой он отличается от предшествующего набора. Например, указанный выше код Грея можно коротко записать в виде последовательности семи чисел 1, 2, 1, 3, 1, 2, 1. В общем случае длина подобной последовательности равна $2^n - 1$. Указанная краткая запись позволяет догадаться, как можно строить коды Грея дальше. Например, Грея

порядка 4 можно задать последовательностью 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1. Она получается, если мы повторим два раза последовательность, определяющую код Грея порядка 3, разделив оба экземпляра этой последовательности числом 4. Далее поступаем аналогично, т.е. последовательность длины $2^n - 1$, определяющую код Грея порядка n , дублируем, разделив оба дубля числом $n + 1$. Полученная последовательность длины $2(2^n - 1) + 1 = 2^{n+1} - 1$ будет определять код Грея порядка $n + 1$.

Легко видеть, что эта последовательность, так же, как и предыдущая, не изменяется, если ее выписать в обратном порядке. Последовательности (или слова) с таким свойством называются палиндромами. Например, одним из самых известных русских палиндромов является предложение (символы пробела, естественно, игнорируются)

А РОЗА УПАЛА НА ЛАПУ АЗОРА.

Нам понадобится это свойство кода Грея порядка $n + 1$, а также то обстоятельство, что все числа этой последовательности, кроме среднего, не больше n . Это означает, что если с ее помощью мы начнем выписывать последовательность наборов длины $n + 1$, начиная с нулевого, то первые $2^n - 1$ наборов будут начинаться с нуля, так как $(n + 1)$ -я компонента никогда в них не будет меняться. Остальные же компоненты будут образовывать наборы длины n , и они будут чередоваться в том же порядке, что и в коде Грея порядка n , поэтому получится некоторая перестановка всех $2n$ наборов длины $n + 1$, начинающихся с нуля. Потом в ее последнем наборе этот ноль будет заменен на единицу (это определяется тем, что в середине последовательности стоит число $n + 1$), далее эта единица меняться не будет, а будет меняться в каждом очередном наборе длины $n + 1$ только одна из последних n компонент, и эти компоненты образуют код Грея порядка n , выписанный в обратном порядке, и закончится он нулевым набором. Сама же построенная при этом последовательность наборов длины $n + 1$ будет образовывать перестановку всех наборов длины $n + 1$, начинающихся с единицы, а ее последним набором будет набор 10...0. Таким образом, построенная последовательность состоит из 2^n различных наборов и может быть превращена в циклическую, т.е. является кодом Грея порядка $n + 1$.

Код Грея можно наглядно изобразить на n -мерном двоичном кубе. Сам этот куб служит для наглядного представления множества всех наборов длины n из нулей и единиц. Наборы изображаются точками и называются вершинами куба. Два набора, отличающиеся только в одной компоненте, называются соседними и образуют ребро куба. Номер этой компоненты называется направлением ребра. Куб можно нарисовать на плоскости так, что

⁵ Приблизительный западный аналог — это конец света, но на Востоке считают, что после этого рождается новый Брахма, и все циклически повторяется сначала: от золотого века до нашей Калли-Юги, которой и заканчивается цикл.

все ребра будут изображены отрезками, соединяющими их вершины, причем ребра одного направления будут изображены равными и параллельными отрезками (поэтому такие ребра тоже называют параллельными). Например, четырехмерный куб можно изобразить на плоскости так, как показано на рис. 3.

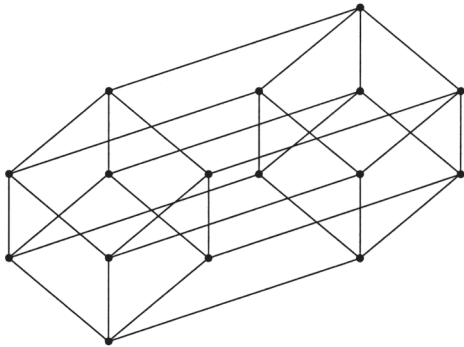


Рис. 3

Код Грея на многомерном кубе можно изобразить в виде последовательности вершин, в которой каждые две соседние вершины соединяются ребрами. Такие последовательности вершин принято называть путями. Но код Грея изображается путем, у которого первая и последняя вершины тоже соединяются ребром. Такие пути естественно называть циклами. Однако код Грея — не просто цикл, а цикл, проходящий через все вершины куба. Такие циклы (а их можно искать не только на многомерном кубе, но и на любом графе⁶) называются *гамильтоновыми циклами*⁷, а графы, у которых они есть, — *гамильтоновыми графами*. Вопрос о том, какие графы гамильтоновы, а какие нет, оказался чрезвычайно трудным и не решен удовлетворительно по сей день. Ему можно посвятить отдельную книгу, и такие книги уже написаны, поэтому мы прекращаем разговор на эту тему и возвращаемся к многомерному кубу.

Для того чтобы изобразить код Грея на приведенном на рис. 3 изображении n -мерного куба, рядом с вершинами следует написать их имена — соответствующие им наборы из нулей и единиц. Сделать это можно, например, таким образом. Самой нижней вершине сопоставим набор из одних нулей. Ребрам

⁶ Граф — это произвольное множество вершин, некоторые из которых соединены ребрами.

⁷ В честь ирландского математика, механика, физика и астронома У.Р. Гамильтона, подсказавшего одному книготорговцу идею головоломки — как обойти по ребрам все вершины правильного многогранника (например, додекаэдра) и вернуться в исходную вершину. Полученные от книготорговца десять фунтов были, вероятно, единственным гонораром знаменитого ученого за многочисленные научные труды, если не считать оклада профессора Дублинского университета.

сопоставим номера их направлений, например, направлению самого правого ребра, выходящего из “нулевой” вершины, сопоставим номер 1, и т.д., а направлению самого левого такого ребра — номер n . Далее сопоставляем оставшимся вершинам куба имена с помощью следующего алгоритма. Если какой-то вершине имя уже присвоено и, поднимаясь из нее вверх по какому-нибудь ребру, скажем, направления k , попадаем в новую, пока безымянную, вершину, то ей присваиваем имя, которое получается из имени прежней вершины заменой k -й компоненты (которая была нулем) на противоположную (т.е. единицу). Если же мы попали в вершину, имя которой уже было присвоено ранее, то можно ничего не делать, так как если мы попробуем все же сопоставить ей имя согласно указанному правилу, то оно совпадет с уже присвоенным именем. Очевидно, самой верхней вершине будет присвоен набор из одних единиц. Результат работы описанного алгоритма для четырехмерного куба показан на рис. 4.

Читателю предоставляется возможность самому решить головоломку Люка и обнаружить ее связь с кодом Грея, а мы займемся другим вопросом.

На изображениях многомерного куба бросается в глаза, что все вершины, имена которых содержат заданное число единиц, скажем k , лежат на одной прямой (рис. 5). Говорят, что эти вершины лежат на k -м слое куба. Очевидно, нулевой слой состоит из одной вершины — “нулевой”, а n -й слой состоит только из “единичной” вершины.

10. Сколько различных “возрастающих” путей ведут из “нулевой” вершины в данную вершину k -го слоя?

Ответ: $k(k-1)(k-2) \cdot \dots \cdot 2 \cdot 1$. Это число называют “ k факториал” и кратко обозначают $k!$.

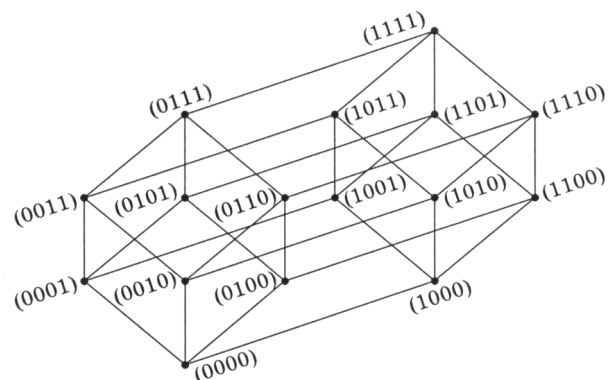


Рис. 4

Для того чтобы решить эту задачу, достаточно “закодировать” любой такой путь последовательно k различных чисел, нумерующих направления составляющих этот путь ребер.

В частности, число возрастающих путей от нулевой вершины до единичной равно $n!$. На любом

таким пути есть единственная вершина, принадлежащая k -му слою, и она разбивает его на две части. Нижняя часть соединяет ее с нулевой вершиной, и этот путь — один из множества возможных путей, соединяющих ее с нулевой вершиной, которых, как мы уже знаем, ровно $k!$. Верхняя часть соединяет ее с единичной вершиной, и этот путь — один из множества возможных, которых, как легко понять по аналогии, в точности $(n - k)!$.

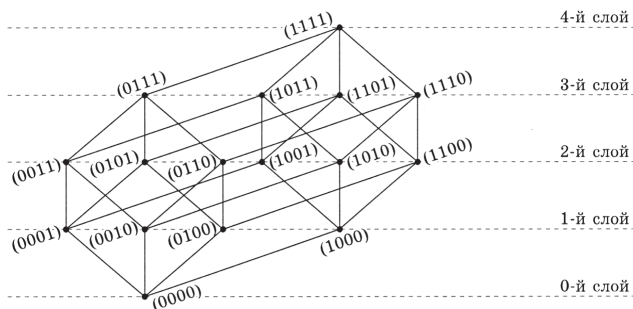


Рис. 5

Поэтому множество всех возрастающих путей от нулевой вершины до единичной, проходящих при этом через заданную вершину k -го слоя, равно $k!(n - k)!$. Но всего возрастающих путей от нулевой вершины до единичной $n!$, поэтому число вершин в k -м слое равно $\frac{n!}{k!(n - k)!}$. Это число называется k -м биноми-

альным коэффициентом и обозначается кратко C_n^k . Далее оно нам понадобится.

На этом мы заканчиваем рассказ о n -мерном кубе и возвращаемся в заключение опять к коду Грея. Очевидно, что в нем несоседние вершины могут быть соседними на кубе, т.е. соединяться в нем ребром. В некоторых приложениях, как теоретических, так и практических, представляет интерес построение цепи или цикла в n -мерном кубе, в которой несоседние вершины никогда не являются соседними в этом кубе. Такая цепь максимальной возможной длины называется “змеей в ящике”. Какова ее длина, до сих пор неизвестно. Наилучшие известные ее оценки принадлежат новосибирским математикам А.А. Евдокимову и А.Д. Коршунову.

Другой задачей, связанной с кодом Грея, но более простой, является его не двоичное обобщение. Например, как построить циклическую последовательность всех трехзначных десятичных чисел от 000 до 999, в которой каждые два соседних числа были бы соседними еще и в том смысле, что отличались бы ровно в одном разряде и ровно на единицу? Несколько проще построить такую последовательность, если считать цифры 0 и 9 также соседними, хотя

разность между ними и не равна 1. Еще проще построить нециклическую последовательность с теми же свойствами.

Эта задача имеет приложение к алгоритму быстрого вскрытия кодового замка на дипломате, поэтому мы не приводим ее решения.

Заметим, что этим методом можно вскрыть, конечно, дипломат не только с десятичным, но и с любым k -ичным кодовым замком. Однако при нечетном k аналога циклического кода Грея не существует, а существует только “цепной” код Грея. Читателю предоставляется возможность самому это проверить.

§ 7. “Книга перемен”, азбука Морзе, шрифт Брайля и алфавитные коды

Двоичная система, по крайней мере в своей комбинаторной ипостаси, по существу была известна в Древнем Китае. В классической книге “И цзин” (“Книга перемен”) приведены так называемые “гексаграммы Фу-си”, первая из которых имеет вид $\equiv \equiv \equiv$, а последняя (64-я) — вид $\equiv \equiv \equiv$, причем они расположены по кругу и занумерованы в точном соответствии с двоичной системой (нулям и единицам соответствуют сплошные и прерывистые линии). Китайцы не поленились придумать для этих диаграмм специальные иероглифы и названия (например, первая из них называлась “кунь”, а последняя — “цян”ь”, сплошной линии сопоставляется мужское начало янь, а прерывистой линии — женское начало инь).

Каждая гексаграмма состоит из двух триграмм (верхней и нижней), им тоже соответствуют определенные иероглифы и названия. Например, триграмме из трех сплошных линий сопоставлен образ-атрибут “небо, творчество”, а триграмме из трех прерывистых линий сопоставлен образ-атрибут “земля, податливость, восприимчивость”. Их также принято располагать циклически, но этот цикл не является кодом Грея.

“Книга перемен” очень древняя, возможно, одна из древнейших в мире, и кто ее написал — неизвестно. Она использовалась ранее, и используется в настоящее время, в том числе и на Западе, для гадания. В Европе с аналогичной целью используются карты Таро. В чем-то обе эти системы схожи, но Таро никак не связаны с двоичной системой, поэтому о них мы говорить не будем.

Способ гадания по “Книге перемен” в кратком изложении таков. Бросается шесть раз монета (или лучше пуговица, деньги в гадании применять не рекомендуется), и по полученным результатам (орел или решка) разыскивается подходящая гексаграмма (для этого надо заранее сопоставить орлу и решке янь или инь). По гексаграмме разыскиваете со-

ответствующий раздел “Книги перемен” (имеется перевод выдающегося синолога Ю.К. Шущкого, неоднократно переиздававшийся в последнее время) и читаете, что там написано.

Конечно, перевод текста книги в предсказание требует опыта и мастерства. И заниматься этим надо после соответствующей подготовки, в подходящем настроении, в подходящее время и в подходящем месте. Говорят, тогда предсказания почти всегда сбываются. А может быть, просто магическим образом из множества вариантов будущего выбирается тот, который соответствует предсказанию?

Заинтересованного читателя отсылаем к книгам Дж.Х. Бреннана. Таинственный И цзин. М.: Гранд, 2001; В.Фирсова. Книга перемен. Мистика и магия Древнего Китая. М.: Центрполиграф, 2002, и переходим к другой теме — азбуке для слепых.

На этом примере, в частности, хорошо видно, что многие на первый взгляд простые идеи рождались не сразу, и своим появлением обязаны усилиям многих людей. Идею использовать рельефные буквы для печатания книг для слепых первым предложил француз Валентен Ойи. Но выпущенные им книги успехом не пользовались, так как слепым трудно было на ощупь отличать сложные начертания букв друг от друга.

Капитан французской армии Шарль Барбье в 1819 году предложил печатать выпуклыми не буквы, а точки и тире (или просто продавливать их на бумаге) и ими уже записывать буквы. Эту систему он назвал “ночное письмо” и предлагал вовсе не для слепых, а для использования в военно-полевых условиях. С появлением электрических фонариков военное значение этого изобретения упало до нуля.

Слепой мальчик Луи Брайль познакомился с этой системой в 12 лет. Она ему понравилась тем, что позволяла не только читать, но и писать. В течение трех лет он ее усовершенствовал и создал так называемый “шрифт Брайля”. В нем символы языка (буквы, знаки препинания и цифры) кодируются комбинациями от одной до шести выпуклых точек, расположенных в виде таблицы стандартного размера с тремя строчками и двумя столбцами. Элементы (точки) таблицы нумеруются числами 1, 2, 3 в первом столбце сверху вниз и 4, 5, 6 во втором столбце сверху вниз. Каждая точка либо продавливается специальной машинкой (или даже шилом), либо остается целой. Всего различных способов продавить выпуклые точки в этой таблице 64 (в том числе и тот, в котором ни одна из точек не вдавлена). При желании теперь читатель может сопоставить каждому символу алфавита Брайля одну из гексаграмм “Книги перемен”. Вряд ли, конечно, Брайль знал об этой книге.

Вероятно, не имеет смысла описывать все символы шрифта Брайля, тем более что после его смерти в 1852 году шрифт дополнялся и совершенствовался.

Но несколько слов сказать, видимо, стоит. Буква “а”, например, изображается одной выпуклой точкой в 1-м элементе таблицы, буква “б” изображается выпуклыми точками в 1-м и 2-м элементах таблицы и т.д. Для сокращения текстов некоторые часто встречающиеся слова или комбинации букв кодируются специальными таблицами. Для того чтобы отличать заглавную букву от строчной, перед ней ставят специальную таблицу, изображающую то, что сейчас называют эскейп-символом. Многие таблицы имеют несколько значений (например, буква и какой-нибудь специальный знак или знак препинания), выбор из которых делается в соответствии с контекстом. Цифры кодируются так же, как и первые буквы алфавита, и, чтобы их отличать, перед последовательностью цифр ставится специальный символ — признак числа, а заканчивается число символом отмены признака числа.

Азбука Брайля по известности уступает азбуке Морзе, хотя и применяется до сих пор в отличие от последней. Сэмюэль Морзе известен, однако, не только изобретением азбуки. Он был и художником-портретистом (его картина “Генерал Лафайет” до сих пор висит в нью-йоркском Сити-Холле⁸), и одним из первых фотографов в Америке (учился делать дагерротипные фотографии у самого Луи Дагерра), и политиком (он баллотировался в 1836 году на пост мэра Нью-Йорка), но самое главное его достижение — изобретение телеграфа (а азбука Морзе понадобилась ему для использования телеграфа). Заодно он изобрел устройство, которое называется реле. Именно из реле спустя сто лет после Морзе были построены первые компьютеры.

Начал свои работы в этом направлении он в 1832 году, запатентовал свое изобретение в 1836 году, но публичная демонстрация телеграфа произошла только 24 мая 1844 года. По телеграфной линии, соединяющей Вашингтон с Балтимором, была успешно передана фраза из Библии.

Точки и тире оказались самыми элементарными символами, которые мог передавать его телеграф. Они соответствовали коротким и длинным импульсам электрического тока, передаваемым по телеграфным проводам. Длина импульса определялась нажатием руки телеграфиста на ключ телеграфа. Прием сигнала осуществляло реле, которое после появления в нем импульса тока включало электромагнит, который либо заставлял стучать молоточек, либо прижимал колесико с красящей лентой к бумажной ленте, на которой отпечатывалась либо точка, либо тире в зависимости от длины импульса.

⁸ Известна также его картина “Человек в предсмертной агонии”, после просмотра которой его приятель, известный врач, сказал: “По-моему, малярия”.

Азбука Морзе сопоставляет каждой букве алфавита последовательность из точек и тире. Естественней всего использовать такие последовательности длины 6, их всего 64 и хватит даже на русский алфавит. Но Морзе понимал, что длину сообщения желательнее уменьшить, насколько возможно, поэтому он решил использовать последовательности длины не более 4, их всего $2 + 4 + 8 + 16 = 30$. В русском алфавите пришлось не использовать буквы “э” и “ё” и отождествить мягкий и твердый знаки. Кроме того, наиболее часто используемым буквам он предложил давать самые короткие коды, чтобы уменьшить среднюю длину передаваемого сообщения. Эту идею в наше время используют с той же целью в алфавитном кодировании.

Здесь имеет смысл ввести терминологию теории кодирования. Определение алфавитного кодирования очень просто. Пусть, например, кодирующим алфавитом является двухбуквенный алфавит, например, состоящий из символов 0, 1. Схемой алфавитного кодирования называется отображение каждой буквы кодируемого алфавита в некоторое слово в кодирующем алфавите (называемое элементарным кодом), в рассматриваемом случае — последовательность нулей или единиц. Пользуясь этой схемой, можно закодировать любое слово в кодируемом алфавите, заменяя в нем каждую букву на соответствующий ей элементарный код, и превратить исходное слово в более длинное слово в кодирующем алфавите. Таким образом, и код Брайля, и азбука Морзе являются алфавитными кодами.

Удобнее всего задать код Морзе в виде четырехярусного двоичного дерева. Из корня дерева выходят два ребра, из которых правому соответствует тире, а левому — точка. Это — ребра первого яруса. Из их концов тоже аналогичным образом выходят по два ребра. Это — ребра второго яруса. Дерево рисуем до четвертого яруса. Вершинам дерева (за исключением корня) приписываем буквы алфавита (рис. 6). Тогда каждой букве можно сопоставить последовательность точек или тире, получающуюся, если выписать друг за другом последовательность символов, сопоставленных ребрам дерева, образую-

щим путь, идущий из корня к вершине дерева, соответствующей данной букве.

Очевидно, что алфавитный код должен обладать свойством однозначной декодируемости, т.е. разные слова в исходном алфавите не могут иметь одинаковые коды. А так как процедуру декодирования можно представлять как поиск разбиения закодированного слова на элементарные коды, то это разбиение должно быть однозначным. Поэтому однозначно декодируемые коды иногда называют разделимыми кодами. Ясно, что если все элементарные коды имеют одинаковую длину, то код разделим и алгоритм декодирования очень прост. Но если это не так, то код может и не быть разделимым. Например, таким является код Морзе. Но между словами телеграфисты всегда делали промежутки, поэтому никаких проблем не возникало. Однако если промежутки между словами выделять невозможно, приходится использовать только разделимые коды. А пустую букву, изображающую промежуток между словами, часто включают в состав алфавита, что дает возможность всегда иметь дело не с предложением, а только со словами.

Понять по достаточно сложному алфавитному коду, является ли он разделимым, бывает не просто. Известны несколько разных алгоритмов для проверки разделимости кода. Наиболее наглядный из них принадлежит А.А. Маркову⁹.

Не вдаваясь в подробности, ограничимся замечанием, что код, у которого ни один из элементарных кодов не является началом другого (такие коды принято называть *префиксными*), несомненно, является разделимым. Предоставим доказательство этой теоремы читателю. Очевидным примером префиксного кода является любой код с равными длинами кодовых слов. Код Морзе префиксным не является, что также предоставляется проверить читателю. Любой двоичный префиксный код можно задать подобно коду Морзе с помощью двоичного дерева, не обязательно такого равномерного, как у него, но

⁹ Заинтересовавшегося этими вопросами читателя мы отсылаем к книге А.А. Маркова “Теория алгоритмов”. М.: Наука, 1984; М.: Фазис, 1996.

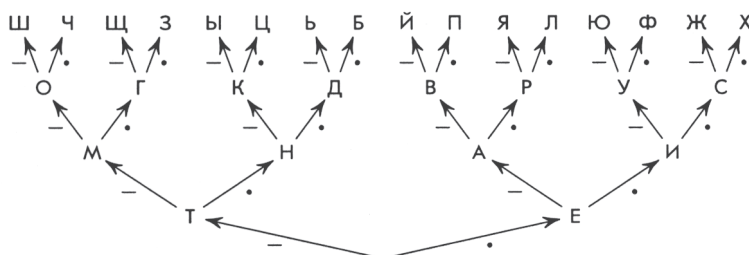


Рис. 6

при этом буквы кодируемого алфавита должны сопоставляться только “листьям” дерева, но никак не внутренним вершинам.

Возвращаясь к истории алфавитного кодирования, заметим, что его корни уходят в глубь веков. Фактически первый пример применения алфавитного кодирования был описан древнегреческим историком Полибием. Алфавит записывался в квадратную таблицу 5×5 , и каждая буква шифровалась парой своих координат (i, j) (номера строки и столбца), а передаваться сообщения могли в то время с помощью факелов — i факелов в левой руке и j факелов в правой означали пару (i, j) . Дальнейшее развитие идеи алфавитного кодирования принадлежит знаменитому английскому философу, эзотерику и писателю сэру Френсису Бэкону, который первым начал использовать двоичный алфавит в качестве шифроалфавита. В криптографии, правда, это не нашло особого применения, главным образом из-за пятикратного удлинения шифртекста в сравнении с открытым текстом. Но сам Бэкон предложил использовать его как метод, сочетающий криптографию со стеганографией (так называется скрытие самого факта передачи секретного сообщения).

Вместо двоичных цифр он использовал обычный алфавит, но со шрифтами двух типов. Таким методом можно было в любом тексте спрятать шифровку, если, конечно, шрифты были достаточно мало-различимы. Желательно при этом использовать разделимый код. Длина зашифрованного сообщения будет в несколько раз короче, чем длина содержащего его (и одновременно маскирующего его) текста, но если для передачи шифровки использовать книгу, то в ней можно таким образом незаметно разместить еще целую книгу. Но эта красивая идея из-за дороговизны ее реализации так и не нашла применения. В наше же время ее нельзя рассматривать как серьезный метод.

Интересно, что в XIX веке, главным образом в кругах, интересующихся наследием возникшего в средневековье тайного мистического ордена розенкрейцеров, появилась идея, что Френсис Бэкон, которого считали розенкрейцером, является настоящим автором пьес Шекспира. Начали искать подтверждение этого в шифрах, которые мог оставить Бэкон в своих книгах, а также в первом знаменитом издании пьес Шекспира. Было, естественно, найдено много таких, якобы зашифрованных фрагментов. Серьезные исследователи, правда, замечали, что в любом длинном тексте можно при желании и некоторых натяжках найти короткие фрагменты, напоминающие шифры. Но у сторонников авторства Бэкона стремление доказать это криптографическим методом приняло форму мании.

Американский миллионер Фабиан даже создал в начале XX века на свои деньги лабораторию криптоанализа, которая занималась только подобными исследованиями.

Фабиан нанял на работу дипломированного генетика Уильяма Фридмана, сына эмигрантов из России. Через некоторое время Фридман уже возглавлял у Фабиана и лабораторию генетики, и лабораторию криптоанализа. Доказать авторство Бэкона он не смог, более того, он впоследствии опубликовал книгу, где опровергал возможность такого криптографического доказательства. Но он не на шутку увлекся криптографией и своей подчиненной Элизабет Смит, с которой обвенчался в 1917 году. Они стали самой знаменитой супружеской парой в истории криптографии. После вступления Америки в войну у него с супругой появилась серьезная работа по правительственным заказам. После войны он ушел от Фабиана и стал главным криптографом войск связи.

§ 8. Фотопленка и штрих-код

Рассмотрим теперь некоторые примеры реального применения двоичного кодирования в современной технике.

Как автоматические фотоаппараты узнают светочувствительность заправленной в них пленки? Ее измеряют в некоторых единицах, и вся выпускаемая сейчас в мире пленка имеет одно из 24 стандартных значений светочувствительности. Эти значения кодируются некоторым стандартным образом наборами из нулей и единиц, естественно, длины 5. На поверхности кассеты для пленки нанесены 12 квадратиков черного или серебристого цвета, образующих прямоугольник 2×6 . Квадратики его верхней части мысленно занумеруем от 1 до 6, начиная слева. Квадратики нижней части аналогично занумеруем от 7 до 12. Серебристые квадратики — это просто металлическая поверхность кассеты, она проводит ток, который с контакта внутри аппарата подается на первый квадрат (он всегда серебристый). Черные квадраты покрыты краской, не проводящей ток.

Когда пленка вставляется в аппарат, шесть его контактов соприкасаются с шестью первыми квадратиками, и с квадратиков со 2-го по 6-й снимается информация — ноль, если квадратик черный и ток по соответствующему контакту не идет, и единица в противном случае. Вся информация о светочувствительности пленки заключена в квадратиках со 2-го по 6-й. В остальных квадратиках заключена информация о числе кадров в пленке и т.п.

Еще на поверхности кассеты можно увидеть штрих-код. Это так называемый “универсальный код продукта”, он сейчас ставится на всех про-

даваемых товарах. Для чего он нужен и как его прочесть?

Нужен он только для автоматического занесения информации в кассовый аппарат. Сам штрих-код состоит из тридцати черных полос переменной толщины, разделенной промежутками тоже переменной толщины. Толщина полос может принимать четыре значения — от самой тонкой до самой толстой. Такую же толщину могут иметь и промежутки. Когда по сканеру проводят штрих-кодом, он воспринимает каждую черную полосу как последовательность единиц длины от одной до четырех и также воспринимает промежутки между полосами, но при этом вместо единиц сканер видит нули. Полностью весь штрих-код сканер воспринимает как последовательность из 95 цифр 0 или 1 (их давно уже принято называть битами). Что же содержит этот код? Он кодирует 13-разрядное десятичное число, совершенно открыто написанное под самим штрих-кодом. Если сканер не смог распознать штрих-код, то это число кассир вводит в аппарат вручную. Штрих-код нужен лишь для облегчения распознавания сканером изображения. Распознавать цифры, к тому же повернутые боком, может только сложная программа распознавания на универсальном компьютере, да и то не очень надежно, а не кассовый аппарат.

Какую же информацию содержит это 13-значное число? Этот вопрос к математике никакого отношения не имеет. Первые две цифры задают страну — производителя товара. Следующие пять цифр — это код производителя, а следующие пять цифр — код самого продукта в принятой этим производителем кодировке. Последняя цифра — это код проверки. Он однозначно вычисляется по предыдущим 12 цифрам следующим образом. Нужно сложить все цифры с нечетными номерами, утроить сумму, к ней прибавить сумму оставшихся цифр, а полученный результат вычесть из ближайшего (большого) кратного 10 числа.

А вот 95-битный код, соответствующий штрих-коду, более интересен. Он содержит в себе только указанное 12-значное число (контрольная цифра в самом штрих-коде не содержится), но с большой избыточностью. Первые три бита в нем, так же, как и последние, — это всегда 101. Они нужны только для того, чтобы сканер смог определить ширину полосы, соответствующей одному биту (ведь размеры штрих-кода на разных упаковках могут быть разными) и настроиться на распознавание. В центре кода всегда стоит комбинация

01010, а левая и правая части кода состоят каждая из шести блоков по семь битов и содержат информацию о левых шести и правых шести из данных 12 десятичных цифр. Центральная комбинация позволяет, в частности, отличать поддельные или плохо напечатанные коды.

Цифры 13-значного кода кодируются в левой и правой частях штрих-кода по-разному. В левой половине каждая цифра кодируется семеркой битов, начинающейся с 0 и заканчивающейся 1 согласно следующей таблице:

■■=0001101=0,	■ =0111101=3,	■■=0111011=7,
■ =0011001=1,	=0100011=4,	■■=0110111=8,
=0010011=2,	■ =0110001=5,	=0001011=9.
	=0101111=6,	

В правой половине каждая цифра кодируется семеркой битов, начинающейся с 1 и заканчивающейся 0 согласно таблице, которая получается из вышеприведенной, если в ней нули заменить на единицы и единицы на нули (это переход к дополнительному коду). Можно заметить, что каждый из кодов в таблице содержит нечетное число единиц и ровно две группы рядом стоящих единиц и ровно две группы рядом стоящих нулей. Это означает, что каждая цифра соответствует двум соседним полосам на штрих-коде. Но более важно то обстоятельство, что все десять кодов таблицы, будучи прочитанными не слева направо, а справа налево, будут отличаться от любого из кодов таблицы, прочитанного правильным образом. Очевидно, таблица для правой половины кода обладает теми же свойствами, только число единиц в каждом коде четное.

Такая избыточная (не четырехбитовая, а семибитовая) таблица кодов нужна для того, чтобы сканер мог правильно прочесть штрих-код и в случае, когда код направляют в него “вверх ногами”. Как сканер может отличать одно направление от другого? По четности или нечетности числа единиц в первом же прочитанном семибитовом блоке, идущем после комбинации 101. При правильном направлении оно будет нечетным, а при обратном направлении — четным. Перепутать же коды, прочитанные слева, и коды, прочитанные справа, согласно свойству таблицы, невозможно.

Если же в каком-то из семибитовых блоков нарушено правильное чередование нулей и единиц в первом и последнем битах или ему не соответствует четность числа единиц, то штрих-код признается поддельным или плохо распечатанным.

Продолжение в следующем номере.

НАЧАЛКА

газета-клуб для всех,
кто учит информатике
маленьких детей



№ 3 (1–15 февраля)

Исполнитель Автомат — так ли все просто?

Н.Д. Шумилина,
г. Тверь

Место Автомата в курсе раннего обучения информатике

Исполнители занимают особое место в курсе раннего обучения информатике. Это инструмент освоения понятия “алгоритм”. Алгоритмизация в пропедевтическом курсе, как и в базовом, представляет одну из основных линий курса. Освоение алгоритмических разделов эффективно построить на основе решения задач — составления алгоритмов. И уже в начальной школе можно и необходимо построить систему ознакомительного обучения основным алгоритмическим конструкциям — следованию, ветвлению, циклическим действиям, вспомогательным алгоритмам и даже рекурсии.

Построение системы задач для курса информатики вообще и курса раннего обучения информатике в частности — процесс трудоемкий, но крайне необходимый и важный. При построении последовательности задач важно выполнение принципов:

- постепенное усложнение: от ознакомительных задач до задач повышенной сложности;
- наследование: необходимость использования знаний, полученных при решении предыдущих задач;
- новизна: добавление каждой следующей задачей новых элементов знаний (для младших школьников, например, это и новый сюжет, и иное

количество элементов, особенностей исходной или требуемой информации).

Существуют коллекции задач разного уровня сложности для освоения алгоритмических разделов начального курса обучения информатике. Они основаны на классических сюжетах: это задачи о переправах, переливаниях, разбегах, фальшивых монетах. Эти задачи предоставляют возможность составления линейных, циклических, ветвящихся алгоритмов. Идея вспомогательного алгоритма прекрасно раскрывается с помощью коллекции задач для исполнителя Машинист. А задача о Ханойских башнях давно стала классическим примером ознакомления с рекурсивным способом организации действий.

Исполнитель Автомат (курса раннего обучения информатике “Роботландия”) — один из простейших исполнителей, использующий арифметические операции. Система команд исполнителя проста: он умеет либо прибавлять 1 к имеющемуся числу, либо умножать имеющееся число на 2. В начальный момент на табло Автомата всегда 0. С помощью этого исполнителя решаются задачи на составление алгоритма получения заданного числа за наименьшее количество операций.

Решая задачи для Автомата, школьники осваивают понятие алгоритма (а именно линейного алгоритма), а также возможную форму его записи. При этом происходит обучение реализации созданного алгоритма с помощью компьютера.

Наличие последовательности заданий разного уровня сложности для исполнителя Автомат способствует цели обучения: формированию навыков работы с арифметическими исполнителями и решению задач на составление оптимальных алгоритмов.

Методика изучения

Получение заданного числа любым способом

По методическим рекомендациям (да и по опыту работы) на первом занятии происходит знакомство с исполнителем и предлагается решение нескольких задач.

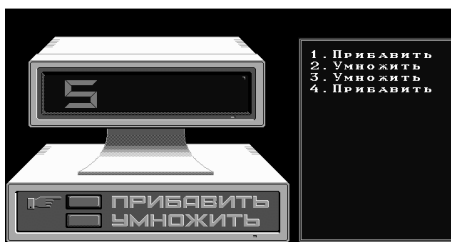
Например, мы хотим получить число 5. Естественно, первой командой может быть только команда “Прибавить”, поскольку в начальный момент на экране Автомата всегда 0. Ход рассуждений фиксируется на доске в виде таблицы:

Команда	На экране	
	До выполнения	После выполнения
Прибавить	0	1

Дальнейший план команд может быть разным: можно получить 5 последовательным применением команды “Прибавить”, т.е. многократным сложением единиц, но есть возможность сократить количество шагов, используя команду умножения на 2. Поскольку пример прост, то применение команды “Умножить” не вызовет затруднений:

Команда	На экране	
	До выполнения	После выполнения
Прибавить	0	1
Умножить	1	2
Умножить	2	4
Прибавить	4	5

Естественно, после разбора и демонстрации работы исполнителя детям предлагается исполнить задание на компьютере самостоятельно. Вид экрана для этого примера показан на рисунке:



Далее дается задание: получить числа 11, 21, 99. Пока перед школьниками задача получения заданного числа за минимальное количество шагов не ставится.

Учитель может фиксировать на доске результаты работы на разных компьютерах, уточняя, что план решения желательно сделать короче.

Опыт работы показывает, что число 11 дети получают за 6 или 7 шагов, на получение числа 21 уходит от 7 до 11 шагов, а для получения 99 разброс велик — некоторые ученики получают это число за 10 или 11 действий, но обычно решения находятся в диапазоне 13–30 шагов.

В конце урока подводится итог, кому удалось получить наиболее короткие решения.

Освоение оптимального (кратчайшего) пути получения заданного числа

На втором занятии рассматривается способ построения оптимального алгоритма. На третьем проверяется домашнее задание, которое должно продемонстрировать умение учащихся применять предложенный оптимальный способ получения заданных чисел.

Работа с Автоматом по оптимальному алгоритму напрямую зависит от уровня математической подготовленности учеников. Перед уроками по освоению Автомата учителю информатики (если именно он, а не учитель начальной школы, в классе которого проходит урок, ведет занятия) просто необходимо проконсультироваться с ведущим учителем класса о сформированности умения школьников делить двузначные числа на 2. Вернее — уточнить диапазон тех чисел, с которыми работа школьников не будет затруднительна.

В книге для чтения № 3 ПМК “Роботландия” подробно описан алгоритм оптимального получения заданного числа с помощью исполнителя Автомат. Он состоит в рассуждении по следующей схеме.

Пока не будет получен 0, повторять следующие рассуждения и действия.

Сначала необходимо ответить на вопрос: “Если на экране число X , то с помощью какой операции (умножения на 2 или сложения с 1) это число было получено?” Если возможны обе операции (число четное), то предпочтение должно отдаваться умножению, так как оно соответствует более быстрому получению заданного числа. Если число нечетное, то оно могло быть получено только операцией сложения с 1.

Далее вычисляется число, предшествующее числу X : либо с помощью операции вычитания 1, либо с помощью операции деления на 2.

Когда достигли числа 0, полученные шаги нумеруются с конца. С одной стороны, получено требу-

емое количество шагов для получения заданного числа. С другой стороны, составлен оптимальный алгоритм получения этого числа. Как пример приведем таблицу, иллюстрирующую анализ получения числа 5:

Экран	Шаги	Действие	Что было
5	4	Прибавить	4
4	3	Умножить	2
2	2	Умножить	1
1	1	Прибавить	0

Столбец “Шаги” заполняется в последнюю очередь, после достижения 0 в столбце “Что было”.

Величина чисел, которые можно предлагать детям в качестве задания, зависит от умения школьников делить на 2. Во всяком случае, деление на 2 чисел, больших 50, может вызывать затруднение учащихся 3-го класса. Поэтому правильный подбор заданий обеспечит усвоение главного — умения анализировать задачу и построение правильного алгоритма ее решения. Иначе арифметические затруднения могут перечеркнуть все усилия по освоению предлагаемого метода.

После объяснения способа получения заданного числа, с одной стороны, необходимо реализовать полученное решение на компьютере, с другой — необходимо применить освоенный способ на других числах.

Выполнение индивидуального задания показывает уровень усвоения материала. Ведь первичное восприятие материала еще не означает его усвоение. А поскольку в существующей школьной действительности очень часто за одним компьютером работают два ученика, то успешная парная работа за компьютером также не является показателем усвоения изучаемого материала каждым ребенком. Если мы хотим и стремимся к качественному обучению, то необходимо в процессе обучения предложить детям полностью самостоятельно решить хотя бы один пример. Результат решения будет показателем либо успешного усвоения, либо затруднения. А затруднение здесь возникает часто в случае проблем с делением числа на 2. В таком случае следует предложить ребенку решить задачу по получению меньшего числа. Поэтому для подобной работы необходим набор упражнений разного уровня сложности.

С этой точки зрения учителю очень удобно иметь распределение чисел (в некотором диапазоне, например, до 100) по количеству шагов, требуемых для получения этих чисел по правилам Автомата.

Количество шагов является показателем сложности решаемой задачи. Множество чисел, получаемых за одно и то же количество действий, позволит подготовить самостоятельную работу для всей группы на одном уровне сложности. Для школьников, быстро выполнивших работу, можно оперативно предложить дополнительное, более сложное задание. Подобный индивидуальный подход не труден, но очень результативен.

Анализ задачи, предшествующий работе на компьютере, очень важен еще и по другой причине. Он демонстрирует детям один из способов решения задач, один из вариантов анализа и приучает к тому, что компьютер — великолепный инструмент и помощник в работе человека, но грамотная работа человека — первична.

Возможные уровни сложности заданий

В таблице представлены числа от 1 до 100. Они распределены по количеству шагов, допустимых для исполнителя Автомат, необходимых для получения этих чисел оптимальным способом.

Количество шагов	Числа	Количество чисел
1	1	1
2	2	1
3	3, 4	2
4	5, 6, 8	3
5	7, 9, 10, 12, 16	5
6	11, 13, 14, 17, 18, 20, 24, 32	8
7	15, 19, 21, 22, 25, 26, 28, 33, 34, 36, 40, 48, 64	13
8	23, 27, 29, 30, 35, 37, 38, 41, 42, 44, 49, 50, 52, 56, 65, 66, 68, 72, 80, 96	20
9	31, 39, 43, 45, 46, 51, 53, 54, 57, 58, 60, 67, 69, 70, 73, 74, 76, 81, 82, 84, 88, 97, 98, 100	24
10	47, 55, 59, 61, 62, 71, 75, 77, 78, 83, 85, 86, 89, 90, 92, 99	16
11	63, 79, 87, 91, 93, 94	6
12	95	1

Приведенная таблица дает возможность составить задания одного уровня сложности в нескольких вариантах в диапазоне от 5 до 11 шагов алгоритма, что фактически означает 7 уровней сложности. Этого вполне достаточно для усвоения главной идеи Автомата и способа построения опти-

мального алгоритма. Более сложные задания, по необходимости, каждый учитель без труда подготовит самостоятельно. Интересная особенность полученных количеств чисел: этот ряд представляет собой не что иное, как числа Фибоначчи (если не ограничиваться верхней границей 100, а рассматривать большие числа).

Важность освоения оптимального решения задач Автомата

Построение оптимального алгоритма для исполнителя Автомат напрямую служит и усвоению понятия “алгоритм”, и одной из возможных форм его записи, и необычному способу построения необходимых действий.

Работа с Автоматом подготавливает школьников к понятию стека. Команды, которые были получены школьником с помощью анализа, записаны на бумаге и представляют стековый способ хранения команд.

С другой стороны, очевидна развивающая сторона Автомата в самом широком смысле: движение мысли от конца к началу — это основа анализа, построения плана решения задач вообще (и, например, плана для решения задач исполнителя Машинист). На этом построен метод последовательного уточнения или метод нисходящего проектирования, один из основных методов информатики. Он предполагает движение мысли в двух направлениях: прямом и обратном. В обратном — при составлении плана решения. В прямом — при реализации составленного плана. Д.Пойа указывает

ет: “Продвижение от конца к началу — это общий и полезный метод составления плана”. Освоение этого метода — не простой и не скорый процесс. И Автомат — именно тот исполнитель, который явно демонстрирует обратный ход мысли, а также формирует умение анализировать. Важность формирования такого умения трудно переоценить.

Использованная литература

1. Лапчик М.П., Семакин И.Г., Хеннер Е.К. Методика преподавания информатики: Учебное пособие для студентов педагогических вузов. М.: Издательский центр “Академия”, 2001.
2. Первин Ю.А. ПМК по начальной школе. Роботландия. Книга для чтения, тетрадь № 3. М.: КУДИЦ, 1993.
3. Первин Ю.А. Методика раннего обучения информатике: Методическое пособие. М.: БИНОМ. Лаборатория знаний, 2005.
4. Пойа Д. Математическое открытие. М.: Наука, 1976.
5. Шумилина Н.Д. Переправа, переправа... Задачи разного уровня сложности. // Информатика в школе: Приложение к журналу “Информатика и образование” № 6, 2003.
6. Шумилина Н.Д. Переливашка и Водолей. Задачи разного уровня сложности. // Информатика в школе: Приложение к журналу “Информатика и образование” № 3, 2004.
7. Шумилина Н.Д. Машинист. Задачи разного уровня сложности. // Информатика в школе: Приложение к журналу “Информатика и образование” № 7, 2004.

Компьютерная компетентность в контексте развития личности ребенка (из опыта работы Филимоновской школы)

О.Ю. Кондратьева, Н.Г. Жукова, учителя начальных классов

Филимоновская школа, которой чуть больше 10 лет, расположена в селе Филимоново Переславского района Ярославской области. Эта совсем небольшая (30 учеников) малокомплектная основная сельская школа стала одной из первых школ в районе, где компьютер и информатика давно и прочно вошли в школьную жизнь. Уроки информатики в этой школе уже более четырех лет ведутся во всех классах — с первого до девятого. Последние два года школа подключена к сети: имеет свой элек-



тронный адрес и свой сайт в Интернете.

За это время школа добилась определенных успехов в информатизации учебного процесса, освоении компьютеров и информационных сетей: призовые места и гранты во всероссийских и международных проектах, участие в дистанционном обучении, учительские семинары по информационным технологиям... Это стало возможным благодаря изначально выбранной стратегии информатизации школьного

учебного процесса. Важно, что эта стратегия формировалась не снизу, а сверху. Широкое использование компьютеров в современной российской школе, допускающее, в частности, применение компьютеров в начальных классах, возможно только в начальном компоненте, то есть в рамках ответственности руководства школы. Можно привести множество примеров, когда отношение директора полностью определяет положение дел с учебным использованием компьютеров.

В этом отношении директор Филимоновской школы, заслуженный учитель Российской Федерации Валентина Ивановна Мостовых, — полное подтверждение этого тезиса: филолог по специальности, она хорошо понимает, что дают современные информационные технологии нынешней школе как в области методики, так и в области управления школой. В классе информатики ее можно увидеть столь же часто, как и в директорском кабинете. В школе создана обстановка, в которой уважительное отношение к компьютеру, информатике, компьютеризованному уроку становится престижным.

В школе разработана (создана директором школы и обсуждена с учительским коллективом) и повешена в учительской большая схема, в которой указаны общие, стратегические цели школы в совершенствовании учебного процесса средствами современных информационных технологий. В схеме отмечено место каждого из 13 учителей школы — те конкретные методические задачи, которые ему в ближайшие годы (еще точнее — к ближайшей переаттестации) предстоит решить. Когда разрабатывался школьный сайт “Мы из Филимоново”, на странице каждого учителя, рядом с его фамилией и портретом, было выписано содержимое соответствующей ячейки этой большой стратегической схемы. Сегодняшнее выступление — это, по существу, рассказ об одной из таких ячеек.

Эта ячейка связана с проблемами информатизации начальной школы. Мы сейчас оставляем в стороне актуальность этого фундаментального и в практическом, и в теоретическом отношении тезиса. Он заслуживает отдельного, подробного и очень серьезного обсуждения.

Компьютер приходит к школьникам разными путями: урок, проекты в рамках кружковых занятий, участие в работе дистанционного обучающего центра... Если говорить о формальной стороне вопроса, то в официальных документах занятия учащихся начальной школы информатикой записаны как кружковые. Но все дети настолько увлечены компьютерными возможностями, что кружок от класса в нашей малокомплектной школе не отличается по составу. В каждом классе такие занятия проходят один раз в неделю.

Особенностью и специфической сложностью компьютеризованных предметных уроков начальной школы (особенно младших, 1-х и 2-х классов) является взаимообусловленность учебного процесса по информатике и предметным урокам: компьютер используется как

инструмент совершенствования навыков чтения и освоения грамоты, хотя для освоения компьютера требуются, в свою очередь, элементарные навыки чтения и письма (клавиатурного набора). Здесь существенную роль играют графические возможности программ, используемых младшими школьниками.

Наиболее важным элементом организации занятий в начальной школе являются отношения классного руководителя и учителя информатики. Мы испытали одну из таких схем, по которой первый год обучения (1-й класс) дети занимаются с учителем информатики. При этом учитель начальной школы присутствует на всех без исключения занятиях детей. Такая организация отношений преследует несколько целей:

- классный руководитель практически и очень конкретно изучает материал компьютеризованных фрагментов урока;
- классный руководитель наблюдает пополнение умений и навыков учащихся, используя эти наблюдения в своей общей работе с классом;
- учащиеся осознают важность занятия, видя активную заинтересованность своего классного руководителя, который для младшего школьника всегда является самым высоким авторитетом.

В течение этого года оба учителя — и классный руководитель, и учитель информатики — обсуждают материал, выносимый на занятия так, чтобы конкретное наполнение урока максимально соответствовало запланированному учебному плану в первом классе.

Эти обсуждения, с одной стороны, упрощаются тем, что основная часть программных продуктов начальной школы сконструирована с требованиями предметных дисциплин и пока не затрагивает чисто информатических понятий — “алгоритм”, “исполнитель”, “система команд” и др. Речь идет о программах, иллюстрирующих уроки с букварем, простейшую арифметику, элементы природоведения... Вместе с тем на этих же занятиях в фоновом режиме осваиваются основные навыки простейшего клавиатурного набора и управления мышкой.

С другой стороны, активно и конструктивно участвуя в выборе используемых на занятиях по информатике программных продуктов, учитель начальной школы знакомится с имеющимся в школе набором программ. У нас он достаточно большой. Это в первую очередь многочисленные программы Роботландии, программы фирмы “Никита”, набор программ “Фантазия”, разработки методического центра “Кирилл и Мефодий”, программный пакет “Малыш” Ассоциации “Компьютер и детство”, программа “Буквария”, пакет “Радуга” и др. Ребята много рисуют на уроках информатики, работая с разнообразными графическими программами — “Колобок” из “Фантазии”, “Маляр” из “Информационной культуры”, “Раскрашка” из “Хитов Роботландии”. Учителя начальной школы органически вплетают в компьютеризованные уроки презент-

тации Power Point, которые позволяют не только излагать материал в заранее продуманной методической последовательности, но и организовывать для детей игры на экранах компьютеров и веселые представления в классе. Так, практически целиком на базе презентаций был проведен у первоклассников интересный открытый урок “Праздник азбуки” с приглашением родителей. Некоторые из родителей впервые увидели своих детей у компьютерных экранов.

Предварительная подготовка учителя в течение всего первого учебного года позволяет классному руководителю с начала второго года обучения включиться в процесс информатического образования детей еще более активно.

Разделение такой подготовки учителя начальной школы к самостоятельному проведению занятий обосновывается, в частности, и тем, что к этому времени (за счет освоения детьми основных интерфейсных навыков) учитель начальной школы получает замечательную возможность сосредоточиться на тематическом содержании урока и не отвлекаться на рассказы детям о клавишах, кнопках, мышке и других инструментально-вспомогательных элементах урока. Дети уже освоили эти навыки, а для учителя начальной школы рассказ о них составлял бы дополнительную напряженность.

Обсудив в том же составе (учитель информатики — классный руководитель) и по согласованию с руководством школы, классный руководитель берет на себя самостоятельное проведение нескольких уроков.

При планировании и подготовке компьютеризованных уроков учителю приходится не только обоснованно выбрать необходимую компьютерную программу, но и для тех программ, которые выполнены в виде наполняемых программных оболочек, задать необходимые величины-параметры, например, диапазон используемых случайных чисел (в арифметических программах пакета “Радуга”), сформировать эталонные тексты для упражнений по русскому языку или математике (в адаптированном учебном текстовом редакторе Микрон) и т.д.

1Справка	2Заполнить символ	3Заполнить строку	4Искать	5Форм
Вычислите разности				
21	-	12	=	?
32	-	17	=	??
61	-	22	=	??
18	-	7	=	??
34	-	11	=	??
53	-	20	=	??
Найдите недостающие слагаемые				
23	+	---	=	34
41	+	---	=	57
---	+	22	=	88
71	+	---	=	98
11	+	---	=	52
---	+	30	=	47

Разумеется, компьютеризованный урок — это не 40 минут работы у компьютерного экрана. Компьютер — эффективный дидактический инструмент, но всего лишь один из многих инструментов. Поэтому все традиционные требования к уроку сохраняются, а необходимые материалы комплектуются не столько из учебника информатики, сколько из предметных учебников. Так, при

подготовке компьютеризованных уроков математики был использован традиционный учебник М.И. Моро, Г.В. Бильтюковой. Вообще говоря, к этому времени классный руководитель (учитель начальной школы) уже подготовлен к тому, чтобы сделать компьютеризованными большую часть своих уроков, однако здесь начинают играть свою роль организационные ограничения: класс информатики — один на всю школу, претендентов много, расписание занятий и кружков плотное. В частности, этим объясняется, что в течение года было проведено только несколько таких уроков классного руководителя, в том числе открытых. Итоги уроков обсуждены с учителем информатики, констатировавшим не только рост уровня знаний учащихся, но и освоение навыков владения компьютером классным руководителем. В обсуждении открытых уроков принимают участие широкий коллектив учителей школы и гости.

Одна из целей компьютеризованных уроков в начальных классах Филимоновской школы — опробовать пути, по которым следует идти от формирования простейших навыков общения ребенка с компьютером через освоение умений и знаний информационной культуры к компьютерной компетентности молодого поколения.

Опыт самостоятельного проведения компьютеризованных уроков систематически накапливается учителями начальных классов Филимоновской школы Жуковой Нурией Галимовной и Кондратьевой Ольгой Юрьевной. В информатической подготовке учителей начальной школы помогали учителя информатики Первин Юрий Абрамович и Жуков Кирилл Геннадьевич. Перенос ответственности за проведение компьютеризованных уроков и уроков информатики у младших школьников с учителей-информатиков на учителей начальной школы — это одна из главных концепций в методике пропедевтического курса информатики проф. Первина Ю.А.

В ближайших планах работы с младшими учениками Филимоновской школы:

- расширение круга учителей, привлекаемых к этой работе;
 - расширение круга используемых программ;
 - увеличение доли компьютеризованных предметных уроков, проводимых в классе информатики.
- В более далекие планы отнесены:
- попытки самостоятельной разработки компьютерных программ, необходимых младшим школьникам, на основе анализа формируемых у детей умений и навыков;
 - наблюдение и анализ методических приемов, основанных на школьной практике компьютеризованных уроков;
 - формирование у младших школьников умений и навыков работы в информационных сетях — электронная почта, поисковые системы, сайтостроение;
 - более глубокое и систематическое освоение основ информатики.

В мир информатики

48 (1–15 февраля)

Газета для пытливых учеников
и их талантливых учителей

Диаграммы

Насси — Шнейдермана

Н.М. Тимофеева,

г. Обнинск Калужской обл.

Вы, конечно, знаете о графическом способе записи алгоритма решения задачи — в виде блок-схемы. Менее известен второй способ такого представления алгоритма — диаграмма Насси — Шнейдермана (ее также называют “диаграммой Насси — Шнейдермана”, “N — S-диаграммой” или “структурограммой”).

В своей статье “Краткая история структурных блок-схем (диаграмм Насси — Шнейдермана)” [1] один из авторов диаграммы Бен Шнейдерман пишет: “Пленительная история и эволюция структурных блок-схем (обычно называемых диаграммами Насси — Шнейдермана, или структурограммами) восходит к 1972 году”. Далее он рассказывает, что впервые подумал о создании своих способов записи алгоритмов во время посещения лекции по структурному программированию, когда еще учился в магистратуре. Ему пришло в голову, что если оператор GOTO¹ не должен использоваться, то так же не нужны и соединительные линии в старых блок-схемах. Пятнадцать минут вычерчивания привели к первым идеям по оформлению следования, ветвления и циклов. Вместе с аспирантом Исааком Насси, в то время более глубоко знавшим принципы структурного программирования, они написали статью “Технологии блок-схем для структурного программирования” [2], в которой описали свои идеи и представили новый вид графической записи алгоритмов. Статья была опубликована в августе 1973 года.

С тех пор N — S-диаграммы широко используются в ряде стран. Например, в Германии их применение при документировании программ обусловлено требованиями государственного стандарта этой страны.

Очевидные преимущества N — S-диаграмм заключаются в:

- наглядности;
- отсутствию соединительных линий со стрелками, что помогает избежать случайных ошибок;

— компактности, т.к. даже относительно длинный алгоритм на языке N—S-диаграмм несложно разместить на одной странице;

— простоте использования.

Диаграммы Насси — Шнейдермана строятся с использованием шести элементарных “строительных блоков”.

1. Блок действия

Как известно, алгоритм состоит из последовательности действий. Блок действия используется для представления отдельного действия алгоритма:

Действие

Например:

Взять яйцо

Два действия представляют собой два блока, следующих один за другим:

Взять яйцо

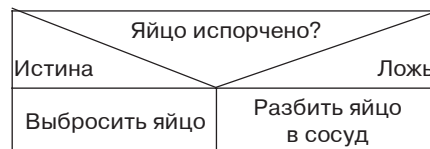
Разбить яйцо

2. Блоки с разветвлением

Блок с разветвлением используется, когда в алгоритме возможны два варианта действий, а выбор того или иного варианта действия зависит от некоторого условия:

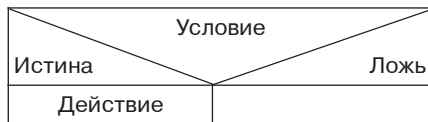


Такая алгоритмическая конструкция (ветвление) представляется двумя смежными блоками действий; действие слева выполняется, если условие верно, действие справа — если условие неверно. Например:



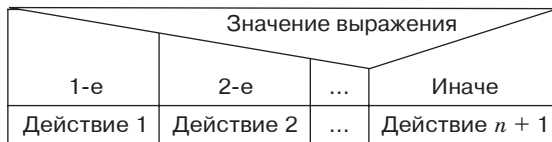
¹ Оператор, использовавшийся в первых языках программирования высокого уровня, при выполнении которого управление передавалось в другое место программы, отмеченное так называемой “меткой”. — Ред.

Возможно также неполное ветвление, при котором некоторое действие выполняется не всегда, а только при определенном условии:



3. Блок множественного выбора

Блок множественного выбора используется, когда существует несколько вариантов возможных действий, выбор которых зависит от значения некоторого выражения²:

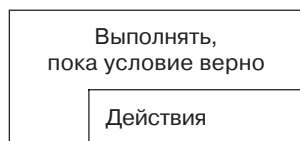


Например, в задаче выбора разных видов обуви для разных видов спорта:

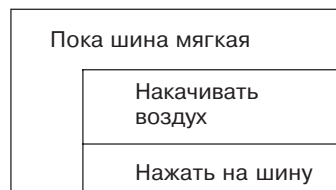


4. Блок цикла с предусловием

Блок цикла с предусловием используется тогда, когда должна быть многократно выполнена некоторая последовательность действий, причем перед каждым выполнением проверяется некоторое условие:



Действия, которые повторяются (так называемое “тело цикла”), представлены самостоятельным блоком внутри блока цикла с предусловием. Например, для задачи: “Накачать спущенную велосипедную шину”:

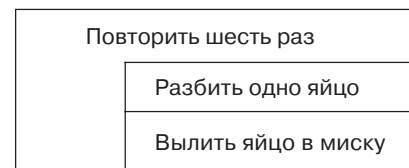


Так как условие проверяется перед выполнением тела цикла, возможно, действия не будут выполне-

² Фрагмент “Иначе” и соответствующий ему фрагмент “Действие $n + 1$ ” в блоке может отсутствовать. — *Ред.*

ны ни разу (если условие ложно в самом начале, вы немедленно выходите из цикла).

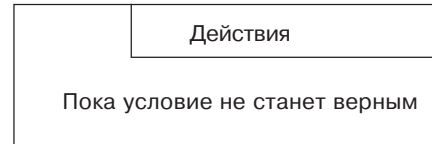
Цикл с заданным количеством повторений тела цикла (в языках программирования его называют “цикл с параметром” или “цикл со счетчиком” [3]) — это тоже цикл с предусловием. Действия повторяются определенное количество раз и отсчитываются *перед* каждым выполнением. Например, разбить в миску шесть яиц:



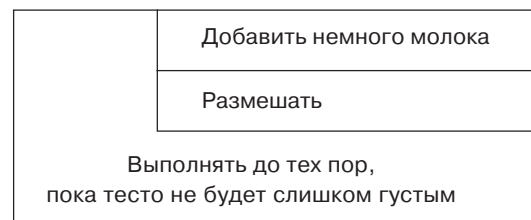
Подсчет действий происходит в начале. Так, если бы вы считали вслух, вы бы сказали “один” перед тем, как разбить первое яйцо.

5. Блок цикла с постусловием

Блок цикла с постусловием используется, когда в алгоритме действия должны повторяться до наступления определенного условия (условие проверяется после выполнения действий):



Например, в задаче приготовления теста для блинов:



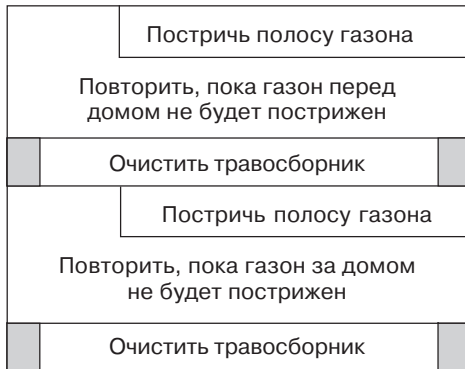
Действия в цикле с постусловием всегда выполняются хотя бы один раз, потому что проверка осуществляется в конце цикла. Так, при приготовлении теста для блинов немного молока будет добавлено и размешано до того, как будет проверена консистенция теста.

6. Блок подпрограммы



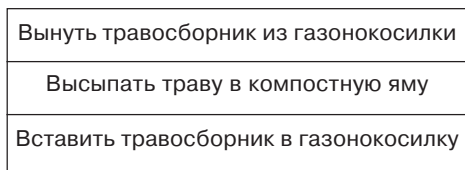
Блок подпрограммы используется в случаях, когда некоторый процесс в алгоритме слишком большой, чтобы изображать его на диаграмме, или когда какие-то блоки действий используются несколько раз

в разных местах одной и той же диаграммы. Например, для задачи стрижки газона около дома диаграмма алгоритма ее решения может быть оформлена так:



Диаграмма, иллюстрирующая действия в подпрограмме, оформляется отдельно.

Подпрограмма “Очистить травосборник”



Перечисленные блоки могут произвольным образом вкладываться один в другой. Проиллюстрируем сказанное на примере задачи: “Найти наименьшее число в следующей последовательности чисел: 51 25 35 79 13 26 65”.

Рассматривая приведенную последовательность чисел, вы можете увидеть, что 13 — наименьшее число в ней, но как вы это определили? Можете ли вы вывести общее правило для определения наименьшего числа в любой последовательности чисел? Есть хорошее упражнение, которое вы можете выполнить с товарищем, чтобы разобраться с этой задачей.

Напишите каждое число на карточке или маленьком кусочке бумаги. Выкладывайте карточки на стол перед вашим товарищем по одной и не забывайте забирать карточку перед тем, как выложите перед ним следующую, т.е. всякий раз ваш товарищ видит только одну карточку.

В конце попросите вашего товарища назвать наименьшее число. Затем сделайте это с любой другой последовательностью чисел. Обсудите с вашим товарищем, что он думал о каждой выложенной карте.

Разработку алгоритма решения задачи и соответствующей диаграммы будем проводить методом, известным под названием “Проектирование сверху вниз”, или “Пошаговая детализация”. Идея метода

очень проста — делим всю задачу на маленькие части и затем рассматриваем каждую часть и детализируем (уточняем) ее до отдельных элементарных действий, которые можно выполнить.

Так как вам придется делать много изменений в ходе работы, стоит вооружиться карандашом, линейкой и ластиком.

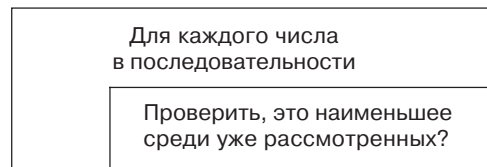
Итак, начнем.

1. Нулевой шаг детализации

Найти наименьшее число в последовательности

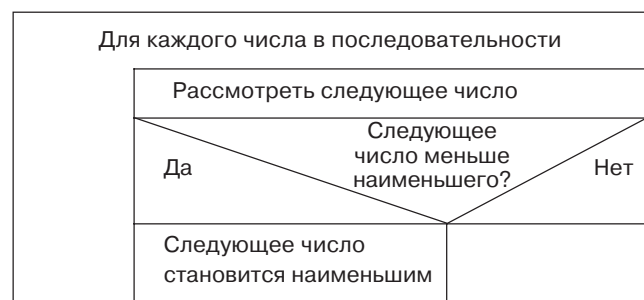
2. Первый шаг детализации

Как вы нашли наименьшее число? Вы просмотрели всю последовательность, рассматривая каждое число и проверяя, не наименьшее ли оно. Это повторяющееся действие — значит, должен быть цикл. Какой? Так как количество повторений известно, то может быть применен цикл с параметром, в N — S-диаграммах являющийся, как указывалось выше, циклом с предусловием:



3. Второй шаг детализации

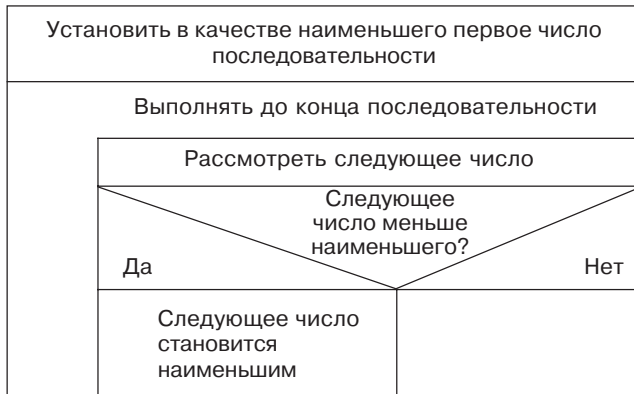
Так что же делать, чтобы понять, какое наименьшее? Детализируем внутренний блок на последней диаграмме. Вот что мы делаем, сравнивая каждое число с наименьшим, найденным до сих пор: если следующее число меньше, оно становится наименьшим, в противном случае старое остается наименьшим³.



4. Третий шаг детализации

Но как вы получили первое наименьшее число? Простейший способ сделать это — принять за наименьшее число первое число в последовательности чисел:

³ Т.е. здесь может быть применен неполный вариант блока ветвления (см. выше). — Ред.



Проверим работу этого алгоритма на примере последовательности чисел: 51 25 35 79 13 26.

1. Установить в качестве наименьшего первое число последовательности (51).
 2. Конец последовательности? — Нет.
 3. Следующее число — 25.
 4. $25 < 51$? — Да.
 5. Устанавливаем в качестве наименьшего число 25.
 6. Конец последовательности? — Нет.
 7. Следующее число — 35.
 8. $35 < 25$? — Нет.
 9. Конец последовательности? — Нет.
 10. Следующее число — 79.
 11. $79 < 25$? — Нет.
 12. Конец последовательности? — Нет.
 13. Следующее число — 13.
 14. $13 < 25$? — Да.
 15. Устанавливаем в качестве наименьшего число 13.
 16. Конец последовательности? — Нет.
 17. Следующее число — 26.
 18. $26 < 13$? — Нет.
 19. Конец последовательности? Да.
- Таким образом, наименьшее число в последовательности равно 13.

Задание для самостоятельной работы

Разработайте диаграммы Насси — Шнейдермана для алгоритмов решения следующих задач.

1. Определить, является ли число A делителем числа B .
2. Даны три целых числа. Определить, сколько из них четных.

Примечание. Циклы не использовать.

3. В чемпионате по футболу команде за выигрыш дается 3 очка, за проигрыш — 0, за ничью — 1. Известно количество очков, полученных командой за игру. Определить словесный результат игры (выигрыш, проигрыш или ничья).

Примечание. Три блока ветвления не использовать.

4. Мастям игральных карт условно присвоены следующие порядковые номера: масти “пики” — 1, ма-

сти “трефы” — 2, масти “бубны” — 3, масти “червы” — 4. По заданному номеру масти m ($1 \leq m \leq 4$) определить название соответствующей масти.

Примечание. Блоки ветвления не использовать.

5. Найти сумму 10 заданных целых чисел.
6. Даны n целых чисел. Определить, сколько из них четных.
7. Известен рост каждого из 20 учеников класса. Рост мальчиков условно задан отрицательными числами. Определить средний рост мальчиков и средний рост девочек.
8. Дано целое число. Если оно положительное, то вычислить средний рост девочек (см. задачу 7), в противном случае вычислить средний рост мальчиков.
9. В задаче 7 выяснить, верно ли, что средний рост мальчиков превышает средний рост девочек более чем на 10 см.

Литература

1. Ben Shneiderman. A short history of structured flowcharts (Nassi-Shneiderman Diagrams). / Draft, May 27, 2003.
2. Nassi I., Shneiderman B. Flowchart Techniques for Structured Programming. / SIGPLAN Notices 8, 8 (August, 1973).
3. Школа программирования: тематический выпуск газеты “В мир информатики”. / “Информатика”, 2005, № 11.

От редакции. Разработанные диаграммы, пожалуйста, присылайте в редакцию. Фамилии всех приславших будут опубликованы, а лучшие ответы мы поощрим.

“Поможем” голову

Кто изображен на портрете?

У Очень-Умникова спросили: “Кто изображен на этом портрете?” Он ответил: “Отец висящего есть единственный сын отца говорящего”. Чей это был портрет?

Задачник

В магазине

Трое мужчин зашли в магазин со своими женами. Каждый из шести купил несколько одинаковых предметов, платя за каждый столько рублей, сколько предметов он купил. Каждый муж потратил на 45 рублей больше, чем его жена; Юрий потратил больше Ольги на 525 рублей, Дмитрий — больше Нины на 13 рублей. Имена остальных — Александр и Татьяна.

Кто на ком женат и сколько предметов куплено каждым?

Блуждания "случайного" жука

Окончание. Начало см. газету-вкладку "В мир информатики" № 66 ("Информатика" № 1/2006)

В этом выпуске рассмотрим методику моделирования процесса перемещения жука по квадрату средствами электронной таблицы Microsoft Excel.

Лист, иллюстрирующий, как меняется картина при одном эксперименте, можно оформить так:

	A	B	C	D	E	F
1	Вероятность=			Всего упал		
2						
3	Эксперимент №1					
4	Длина, см	10				
5	Результат					
6						
7	T	угол	x	y	Упал?	
8	0	0	0	0	-	
9	1	1,10	0,45	0,89		
10	2	2,85	0,50	1,18		
...						
68	60					
69						

В ячейке C1 будет получена искомая вероятность (как именно — опишем ниже).

Формулы, необходимые для расчета положения жука в момент времени t , запишите самостоятельно. При этом следует учесть, что:

1) аргументами функций SIN и COS являются значения, выраженные в радианах;

2) так как в формулах будет использоваться функция СЛЧИС(), то при заполнении новых ячеек значения в ячейках, где использовалась эта функция, будут меняться. Не обращайтесь на это внимание — после заполнения ячеек A8:E68 весь эксперимент можно провести, нажав функциональную клавишу

F9 (при этом происходит перерасчет по всем формулам листа)⁴;

3) ссылка на ячейку B4 должна быть следующей: B\$4.

⁴ Можно также отключить автоматический перерасчет значений на листе — для этого следует в меню программы выбрать пункт Сервис, в нем подпункт Параметры и на вкладке Вычисления установить переключатель рядом с надписью вручную, после чего щелкнуть на кнопке ОК. После этого перерасчет будет происходить только при нажатии клавиши **F9**.

Моделирование

В ячейках столбца E следует получить, например, слово "да", если жук "вышел" за пределы квадрата⁵, и пустую ячейку — в противном случае.

В качестве результата эксперимента, выводимого в ячейке B5, получим 1 или 0 в зависимости от того, упал жук с квадрата или нет. Формула в этой ячейке имеет вид:

=ЕСЛИ(СЧЕТЕСЛИ(E9:E68;"да")>0;1;0)

Итак, одно "путешествие" мы смоделировали. Но, чтобы рассчитать искомую вероятность, таких экспериментов следует провести... 10 000 (!) или хотя бы 9999 — ☺. Оценим наши возможности. Для одного эксперимента требуются 5 столбцов, если значения, соответствующие тексту "всего упал", записывать в ячейке E2. Всего столбцов — 256^6 , значит, мы можем разместить на листе данные и формулы для 51 эксперимента (скопировав имеющиеся 5 столбцов еще 50 раз). При этом первое приближение вероятности (ячейка C1) может быть рассчитано по формуле =E2/51, где значение в ячейке E2 есть общее число экспериментов, в которых насекомое упадет в бак. Это число может быть определено по формуле =СУММ(A5:IU5). Но, конечно, 51 эксперимент — это недостаточно для оценки искомой вероятности. Увеличить число экспериментов можно, используя другие листы (скопировав на них информацию с листа 1). Например, при данных на двух листах формула в ячейке E2 листа 1 должна быть такой: =СУММ(A5:IU5) + СУММ(Лист2!A5:IU5), а в ячейке C1: =E2/(51*2). При увеличении числа используемых листов приведенные формулы меняются соответственно. Однако для моделирования нескольких тысяч экспериментов и 8–10 листов явно недостаточно. Можно поступить следующим образом:

1) оформить рабочую книгу с 10 листами;

2) записать соответствующую формулу в ячейке E2 листа 1;

3) 10 раз нажать функциональную клавишу **F9** (т.е. 10 раз "провести" по 510 экспериментов, смоделированных в рабочей книге);

4) после каждого нажатия записывать значение в ячейке E2 листа 1;

5) сложить (на калькуляторе) все записанные значения.

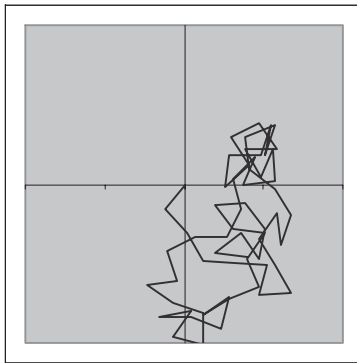
Тогда вероятность падения жука в бак может быть определена как частное от деления полученной суммы на 5100.

⁵ В столбце E возможно появление нескольких значений "да", что, конечно, условно. Можно оформить лист таким образом, чтобы данные не выводились после появления в столбце E первого значения "да".

⁶ Почему именно столько, разъяснено в [1].

Задания для самостоятельной работы

1. Определите вероятность падения жука при $T = 60$ сек.
2. Определите среднее расстояние, на которое удалится *живой* жук от центра через $T = 60$ сек.
3. Получите траекторию движения жука по квадрату:

**Указания по выполнению**

1. С помощью **Мастера диаграмм** постройте диаграмму типа Точечная для значений x и y одного эксперимента.
2. Установите предельные значения для осей — 5 и 5.
3. Уберите основные и промежуточные линии на осях и метки (числа) на них.
4. Проследите, как меняется траектория при каждом новом эксперименте (при каждом нажатии клавиши **F9**).

Ответы и диаграмму также присылайте в редакцию.

Литература

1. Еремин Е.А. 10 познавательных этюдов для Microsoft Excel. Этюд 1. / "В мир информатики" № 16 ("Информатика" № 48/2003).

**Внимание!
Конкурс**

Итоги конкурса № 40 для учащихся

Окончание. Начало см. газету-вкладку "В мир информатики" № 66, 67 ("Информатика" № 1, 2/2006)

Один дурень хвалит себя, другой дурень хвалит сына, самый большой — хвалит жену.

Один испеченный хлеб лучше десяти фунтов теста.

Один конец без другого, голова без хвоста.

Один палец согнешь — все согнутся.

Один пляшет от сытости, а другой от мороза.

Один прилежный стоит тысячи лентяев.

Один раз — победа, другой — поражение: солдату дело привычное.

Один раз раненый всю жизнь боль знает.

Один раз солнце и ад осветит.

Один раз укусит змея — три года будешь бояться пеньковой веревки.

Один ребенок — матери вечные тревоги.

Один ручеек моря не замутит.

Один человек ел ослитину и приговаривал: "Ослиные уши похожи на заячьи".

Один человек роет колодец, пьют воду — тысячи.

Один червяк портит целую кастрюлю похлебки.

Лук всегда имеет *один* запах.

Не стриги бороду при двух людях, ибо *один* скажет "длинна", а другой: "коротка".

Пока *один* абрикос поспевает, сто незрелых осыплются.

**Привели Вера Анкудинова
и Александра Найденова**

Голодной куме *один* хлеб на уме.

На безделье и лень — велик и *один* день, а на хорошие дела — и пятилетка мала.

Один бог святых знает.

Один в соху, да в борону, а другой к черту да в сторону.

Один курит, а весь дом болеет.

Один любит щи наварные, другой жену нарядную.

Сидя у ворот не прокормишь и *один* рот.

Съел ковригу — напитался *один*, прочитал книгу — оставил и другим.

**Прислали Артем Граждан, Игорь Клипов,
Елена Кожаева, Виктория Козлова
и Александр Крадинов**

Эти же ребята привели ряд фразеологизмов:

В *один* голос.

В *один* присест.

Мерить на *один* аршин.

На *один* покрой.

Ни *один* черт.

Все в одного, *один* в *один*.

Один в одного.

Один-другой и обчелся.

Один к одному.

Один конец.

Один на *один*.

Один шаг.

Садиться за *один* стол.

Все как *один*.

За *один*.

Один на другом сидит.

Один на *один* с самим собой.

Один-одинешенек (одна-одинехонька).

Один по одному (одна по одной).

Михаил Чапкевич привел две оригинальные крылатые фразы:

- 1) *один* в четырех каретах;
- 2) *один, один* бедняжечка, как рекрут на часах и турецкие пословицы:
 - за *один* проступок человека не вешают;
 - *один* работает руками, другой — языком;
 - *один* раз в сорок лет жену послушайся;
 - *один* раз каждый ошибается;
 - у муллы пять животов, *один* из них всегда пуст;
 - умный ошибается *один* раз;
 - хитрец, пока не разрушит девять домов, *один* дом не построит.

Всего в конкурсе приняли участие более 60 учащихся⁸. 32 из них прислали более 70 примеров пословиц и поговорок. В результате подведения итогов, при котором учитывалось количество присланных примеров, их соответствие условиям конкурса, оригинальность и т.п., победителями конкурса признаны:

- Абдрахманов Артур;
- Анкудинова Вера и Найденова Александра;

- Баштанова Марина;
- Волынцева Анна;
- Гайзер Яна;
- Граждан Артем, Клипов Игорь, Кожаяева Елена, Козлова Виктория и Крадинов Александр;
- Губанова Марина;
- Исакова Олеся;
- Камалетдинова Лейсан;
- Котельникова Наталия;
- Кривов Дмитрий;
- Кудлацкий Александр;
- Монахова Анна;
- Мошкина Екатерина;
- Пирогова Светлана;
- Погорелова И.;
- Роот Мария;
- Ткаченко Анна;
- Хаустова Кристина;
- Чапкевич Михаил;
- Шамсутдинова Альбина;
- Шнейвейс Анастасия.

Перечисленные учащиеся будут награждены дипломами. Всем остальным участникам конкурса мы говорим: “Спасибо! Вы все — молодцы!”.

Итоги первого тура конкурса № 41

Ответы прислали:

- Абалаков Александр, Глухов Валентин и Полушкин Вадим, Кемлянская средняя школа Ичалковского р-на Республики Мордовия, учитель **Силантьев О.П.**;
- Абдрахманов Артур, г. Учалы, Республика Башкортостан, школа № 2, учитель **Галимова З.Н.**;
- Абдуллин Рамиль, средняя школа деревни Старый Бабиц Кармаскалинского р-на Республики Башкортостан, учитель **Абдуллин Р.Ф.**;
- Агеева Марья, поселок Надвоицы, Республика Карелия, школа № 1, учитель **Богданова Л.М.**;
- Акишина Елена, Боровков Максим, Валиев Ильдар, Латкин Павел, Молчанова Марина, Подавальный Дмитрий и Таранцева Олеся, г. Заозерный Красноярского края, гимназия № 2, учитель **Владимирова Н.А.**;
- Анненков Евгений, Чапкевич Михаил и Якушева Елена, г. Орел, лицей № 4, учитель **Чапкевич И.М.**;
- Антонова Мария, Доронкин Юрий, Егоров Виталий, Зеликова Ольга, Зубащенко Петр, Клепикова

- Светлана, Масыкина Юлия, Мельникова Татьяна, Пименов Дмитрий, Польшванная Елена, Сенчук Светлана и Сушик Василий, г. Новохоперск Воронежской обл., гимназия № 1, учитель **Матыкин В.Ю.**;
- Ахметчина Елена, Балясникова Галина, Жаров Александр, Жезленков Владимир, Найденова Александра и Пудова Наталья, г. Прокопьевск Кемеровской обл., школа № 3, учитель **Шерстнева Л.М.**;
- Ачитуева Арюна, средняя школа села Улюнхан Курумканского р-на, Республика Бурятия, учитель **Доржиева Г.Ю.**;
- Баева Галина, Воробьева Олеся, Зарубина Раиса, Мурашева Екатерина, Тирских Екатерина и Уваровская Валентина, г. Иркутск, школа № 75, учитель **Выборова Е.С.**;
- Бауков Антон и Шарипов Ильфир, основная школа села Васильевка Мелеузовского р-на, Республика Башкортостан, учитель **Зайцева Л.Н.**;
- Баштанова Марина и Чиковинский Дмитрий, г. Братск, лицей № 2, учитель **Коновалова Е.А.**;
- Безсчастная Татьяна и Гладилина Мария, г. Курск, школа № 53, учитель **Пилюгина Е.А.**;
- Богачук Людмила и Размыслова Анна, г. Тюмень, школа № 90, учитель **Размыслова М.В.**;
- Божков Артем, Кононов Владимир, Пушкарев Алексей и Черникова Светлана, средняя школа по-

⁸ Уже перед подписанием номера в печать ответы прислали также Анкудинова Вера и Найденова Александра, г. Прокопьевск Кемеровской обл., школа № 3, учитель **Шерстнева Л.М.**

селка Ивот Дятьковского р-на Брянской обл., учитель **Брин Д.С.**;

— Бомбин Андрей, Кривелев Артем и Малофеев Алексей, поселок Клетня, Брянская обл., школа № 2, учитель **Медведева Л.И.**;

— Бородина Ирина, Голуб Ольга, Губанова Марина, Гурашина Мария, Идиятов Владимир, Косячкова Марина и Муллагалиева Адила, г. Ревда Свердловской обл., школа № 28, учитель **Кольцова Е.М.**;

— Брюханов Алексей, средняя школа села Лугавское Минусинского р-на Красноярского края, учитель **Брюханова Т.А.**;

— Васильев Александр, г. Мелеуз, Республика Башкортостан, гимназия № 1, учитель **Животова Е.П.**;

— Вовченко Дарья, г. Киров, школа № 28, учитель **Дубинина У.В.**;

— Войтик Константин, средняя школа поселка Ольга, Приморский край, учитель **Биктимиров В.М.**;

— Волкова Марина, Нижний Новгород (номер школы и фамилия учителя информатики в письме не указаны);

— Высоцкий Роман, Пономарева Юлия, Тулеуова Айнура и Хрякова Ольга, г. Омск, школа № 110, учитель **Юст Т.Н.**;

— Гавшина Анастасия и Мамочкина Анастасия, г. Белово Кемеровской обл., поселок Краснобродский, школа № 31, учитель **Мамочкина М.В.**;

— Гайсина Галия, Гайсин Рашит и Хлыбов Владимир, г. Уфа, Республика Башкортостан, школа № 18, учитель **Искандарова А.Р.**;

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

— Глуценко Андрей, г. Кореновск Краснодарского края, школа № 19, учитель **Выскребец Т.В.**;

— Гордиенко Е., средняя школа села Гилевка Алтайского края, учитель **Збарах В.Ф.**;

— Гусейнов Эмиль, Котельников Александр, Малуев Артем и Прохоренко Илья, г. Инта, Республика Коми, гимназия № 3, учитель **Беспалько Н.Т.**;

— Дедобаева Нигина, Летова Елена, Муштакова Татьяна, Хаустова Кристина и Щемак Екатерина, Вознесенская основная школа Красногвардейского р-на Оренбургской обл., учитель **Гриднев А.Б.**;

— Диденко Оксана, Евстегнеева Анна, Иванова Наталия, Климентьева Анастасия и Серебренникова Дарья, г. Нижний Новгород, школа № 77, учитель **Занозина Г.В.**;

— Димитриев Василий, Крыжко Сергей, Кузина Марина, Невструев Дмитрий, Родугин Алексей, Свиридова Анастасия, Сорокин Алексей и Суховерхова Ксения, г. Воронеж, лицей № 2, учитель **Моделина Е.А.**;

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

— Долгов Андрей, г. Благовещенск Амурской обл., школа № 16, учитель **Крюченкова Ю.В.** (преподаватель информатики в УПК № 1 **Валюк Т.В.**);

— Дубовицкий Николай, Житникова Анастасия и Кириенко Наталья, г. Новокузнецк Кемеровской обл., гимназия № 44, учитель **Дубовицкая Н.В.**;

— Еремеев Евгений, средняя школа села Плеханово Кунгурского р-на Пермской обл., учитель **Полякова Н.И.**;

— Ерофеева Валентина и Тихонова Ирина, Ярцевская средняя школа Енисейского р-на Красноярского края, учитель **Шашина Н.П.**;

— Пехов Олег, поселок Белый Яр Верхнекетского р-на Томской обл., школа № 1, учитель **Слезко П.С.**;

— Жамков Сергей, Зотов Антон, Поляков Никита и Саморуков Даниэль, Москва, школа № 829, учитель **Шнейдер Е.В.**;

— Загафуранова Айсылу, средняя школа деревни Сейтяк Балтачевского р-на, Республика Башкортостан, учитель **Загафуранова А.Ф.**;

— Зайцев Максим, г. Белово Кемеровской обл., поселок Краснобродский, школа № 31, учитель **Зайцева Л.А.**;

— Занозин Михаил, г. Нижний Новгород, школа № 172, учитель **Занозин Д.А.**;

— Золоторева Ольга, Минеева Светлана, Победимова Светлана и Трунова Елена, г. Новокузнецк, школа № 100, учитель **Папшева В.В.**;

— Иванов Богдан и Стренин Михаил, г. Новокузнецк Кемеровской обл., Городской дворец детского (юношеского) творчества им. Н.К. Крупской, педагог **Костина В.А.**;

— Иванов Максим и Петрова Мария, г. Шарыпово, Красноярский край, школа № 8, учитель **Балина И.А.**;

— Иванова Татьяна и Трошин Андрей, г. Сегежа, Республика Карелия, школа № 5, учитель **Меньшиков В.В.**;

— Иванова Эвелина, г. Бирск, Республика Башкортостан, школа № 1, учитель **Зонова И.В.**;

— Камалетдинова Лейсан, г. Нижнекамск, Республика Татарстан, школа № 27, учитель **Павлова В.Н.**;

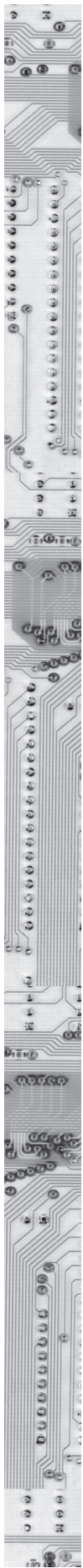
— Капитанович Вячеслав, г. Нижний Новгород, Нижегородское речное училище им. И.П. Кулибина, преподаватель **Занозина Г.В.**;

— Клипов Игорь, г. Балашов Саратовской обл., гуманитарно-педагогический лицей-интернат, учитель **Сухорукова Е.В.**;

— Кожедуб Ирина и Комисарова Регина, г. Мелеуз, Республика Башкортостан, школа № 1, учитель **Иноземцева Н.А.**;

— Коноваленко Вадим, Кудинова Юлия, Поминова Елена и Рязанцев Алексей, г. Воронеж, лицей № 2, учитель **Комбарова С.И.**;

— Кривов Дмитрий, Кудлацкий Александр и Ткаченко Анна, средняя школа села Подсосное Наза-



ровского р-на Красноярского края, учитель **Криво-ва Н.П.**;

— Крупин Федор, г. Ярославль, школа № 33, учи-тель **Ярцева О.В.**;

— Лебакова Татьяна и Чевыкалов Дмитрий, г. Се-веродвинск, школа № 2, учитель **Заборская О.Н.**;

— Любогощев Михаил и Татаркин Максим, г. Ново-кузнецк, гимназия № 44, учитель **Лобко Л.Н.**;

— Митин Илья, г. Зеленокумск Ставропольского края, школа № 3, учитель **Ржевский Д.В.**;

— Молодцов Сергей, Новобурасский р-н Сара-товской обл., Ириновская средняя школа, учитель **Брунов А.С.**;

— Мосарыгин Илья, средняя школа села Кирицы Спасского р-на Рязанской обл., учитель **Мосарыги-на О.В.**;

— Мотелика Артем и Савенко Елена, поселок Лимбяха Новоуренгойского р-на Тюменской обл., школа № 1, учитель **Исакова И.С.**;

— Мухарметова Эльвина, Нафиков Гаяз, Орлов Александр, Сабитова Юлия, Саляхова Неля и Эрш-тейн Антон, г. Стерлитамак, Республика Башкорто-стан, школа № 1, учитель **Орлова Е.В.**;

— Найденова Екатерина и Ратников Евгений, село Спасское Спасского р-на Приморского края, школа № 8, учитель **Мирошник Л.В.**;

— Никитин Вячеслав, средняя школа поселка Пре-дивинск Большемуртинского р-на Красноярского края, учитель **Фиряго И.Н.**;

— Остряков Даниил и Кляубин Тимофей, г. Стар-ый Оскол Белгородской обл., гимназия № 18, учи-тель **Хорунжина И.Н.**;

— Панасенко Илья, средняя школа села Стретен-ка Дальнереченского р-на Приморского края, учи-тель **Панасенко Л.В.**;

— Пивоварова Елена, поселок Сосьва Серовско-го р-на Свердловской обл., школа № 1, учитель **Бед-рина Н.Т.**;

— Пименова Евгения, г. Стерлитамак, Респуб-лика Башкортостан, гимназия № 5, учитель **Пучки-на С.А.**;

— Позднякова Снежана, средняя школа села Ар-хангельское Губкинского р-на Белгородской обл., учи-тель **Позднякова И.В.**;

— Посысаев Роман, г. Бийск Алтайского края, поселок Нефтебаза, школа № 13, учитель **Легя-ва О.В.**;

— Репкина Ева, г. Воркута, Республика Коми, гим-назия № 2, информатику пока не изучает;

— Ростовцев Евгений, г. Тосно Ленинградской обл., гимназия № 2, учитель **Кособокова Ю.С.**;

— Сараев Андрей, средняя школа села Ваганово Октябрьского р-на Челябинской обл., учитель **Сара-ева Н.Н.**;

— Сасимович Андрей, Республика Беларусь, г. Глу-бокое Витебской обл., учитель **Сасимович Е.И.**;

— Семяшкина Екатерина, Республика Коми, г. Сыктывкар, МОУ “Лицей народной дипломатии”, учитель **Гранаткина О.М.**;

— Сигидин Николай, г. Шарыпово, Краснояр-ский край, школа № 8, учитель **Кузьмин П.А.**;

— Силаев Валерий, Ардатовский аграрный тех-никум, поселок Ардатов Нижегородской обл., пре-подаватель **Касаткина С.Ю.**;

— Склюева Юлия, средняя школа села Плехано-во Кунгурского р-на Пермской обл., учитель **Поля-кова Н.И.**;

— Стафеева Елизавета, г. Новоуральск Свердло-вской обл., школа № 58, учитель **Стафеева Н.А.**;

— Тарасов Евгений, Большекалмыкская средняя школа, Киреевский р-н Тульской обл., учитель **Иса-ева Л.В.**;

— Тарасов Юлиан, средняя школа села Зерновое Черемховского р-на Иркутской обл., учитель **Гоня-ева Т.А.**;

— Топоров Сергей и Ткач Николай, поселок Пин-чуга Богучанского р-на Красноярского края, школа № 8, учитель **Зотиков А.М.**;

— Урдиханова Алина, г. Жуков Калужской обл., школа № 2, учитель **Урдиханова И.Н.**;

— Хе Николай и Ющенко Максим, г. Холмск Са-халинской обл., лицей “Надежда”, учитель **Романь-кова С.Н.**;

— Шамсутдинова Альбина, г. Лениногорск, Респуб-лика Татарстан, школа № 8, учитель **Кашапова Р.Х.**;

— Щепин Павел, село Якшур-Бодья, Удмуртия, Якшур-Бодьинская сельская гимназия, учитель **Лес-никова Е.Ю.**;

— Ямаев Ильгизар, поселок Кандры Туймазин-ского р-на, Республика Башкортостан, школа № 2, учитель **Шакирова А.Г.**

Прежде чем представлять решения задач, кото-рые должен выполнить исполнитель Водомер, заме-тим, что при подведении итогов учитывалась опти-мальность присланных решений — правильным счи-тался ответ, в котором задача решается за мини-мальное число операций.

Задача 1

№ пп	Команда	А	В
Исх.		0	0
1	К1	3	0
2	К3	0	3
3	К1	3	3
4	К3	2	4

Задача 2

Эта задача имеет два лучших (принципиально раз-ных) решения, равнозначных с точки зрения числа команд.

Первый способ:

№ пп	Команда	А	В
Исх.		0	0
1	К1	3	0
2	К3	0	3
3	К1	3	3
4	К3	0	6
5	К1	3	6
6	К3	0	9
7	К1	3	9
8	К3	2	10
9	К6	2	0
10	К3	0	2
11	К1	3	2
12	К3	0	5

Второй способ:

№ пп	Команда	А	В
1	К2	0	10
2	К4	3	7
3	К5	0	7
4	К4	3	4
5	К5	0	4
6	К4	3	1
7	К5	0	1
8	К4	1	0
9	К2	1	10
10	К4	3	8
11	К5	0	8
12	К4	3	5

Такие два варианта решения представили Анна Евстегнеева, Николай Дубовицкий, Лейсан Камалетдинова, Ксения Суховерхова, Альбина Шамсутдинова и ребята из гимназии № 2 г. Заозерный. Эти участники конкурса получают за задачу дополнительно 1 балл.

Задача 3

№ пп	Команда	А	В
Исх.		0	0
1	К2	0	10
2	К4	4	6
3	К5	0	6
4	К4	4	2
5	К5	0	2

Задача 4

Здесь также имеются несколько вариантов решения. В лучшем из них задача решается за 6 операций:

№ пп	Команда	А	В	С
Исх.		0	0	0
1	К3	0	0	12
2	К9	0	10	2
3	К1	0	0	2
4	К9	0	2	0
5	К3	0	2	12
6	К9	0	10	4

Такой способ предложили Владимир Кононов, Федор Крупин, Олег Пехов и Наталья Пудова. За это они получают 1 дополнительный балл.

Приведем также два алгоритма решения задачи за 7 операций (таких способов несколько):

1)

№ пп	Команда	А	В	С
Исх.		0	0	0
1	К3	0	0	12
2	К9	0	10	2
3	К1	0	0	2
4	К8	2	0	0
5	К3	2	0	12
6	К9	2	10	2
7	К8	4	10	0

2)

№ пп	Команда	А	В	С
Исх.		0	0	0
1	К3	0	0	12
2	К9	0	10	2
3	К8	2	10	0
4	К1	2	0	0
5	К3	2	0	12
6	К9	2	10	2
7	К8	4	10	0

За решения, аналогичные приведенным, участники конкурса получают 1 балл.

В задачах 5 и 6 оптимальных решений также несколько. Приведем по одному из них.

Задача 5

№ пп	Команда	А	В	С
Исх.		0	0	0
1	К2	0	10	0
2	К7	0	0	10
3	К1	5	0	10
4	К5	3	0	12
5	К2	3	0	0
6	К2	3	10	0
7	К7	3	0	10
8	К5	1	0	12

Задача 6

№ пп	Команда	А	В	С
Исх.		0	0	0
1	2	0	5	0
2	6	4	1	0
3	10	0	1	0
4	6	1	0	0
5	3	1	0	10
6	8	4	0	7

Во многих ответах после получения требуемого объема воды в одной из емкостей использованы команды на выливание воды из других емкостей, что является излишним.

После первого тура с точки зрения количества правильно решенных задач лидерами “чемпионата” являются (в алфавитном порядке фамилий): Абдуллин Рамиль, Абдрахманов Артур, Агеева Мария, Анненков Евгений, Божков Артем, Брюханов Алексей, Вовченко Дарья, Волкова Марина, Высоц-

кий Роман, Гайсин Рашит, Гайсина Галия, Глижинский Дмитрий, Голуб Ольга, Гордиенко Е., Губанова Марина, Дмитриев Юрий, Дубовицкий Николай, Евстегнеева Анна, Ерофеева Валентина и Тихонова Ирина, Жезленков Владимир, Зайцев Максим, Занозин Михаил, Иванов Максим, Иванова Наталия, Иванова Татьяна, Клепикова Светлана, Клипов Игорь, Клюбин Тимофей, Комисарова Регина, Кононов Владимир, Крушин Федор, Кудлацкий Александр, Любогощев Михаил, Масякина Юлия, Мельникова Татьяна, Минеева Светлана, Молодцов Сергей, Мосарыгин Илья, Мурашева Екатерина, Муштакова Татьяна, Найденова Екатерина, Нафиков Гаяз, Остряков Даниил, Петрова Мария, Пехов Олег, Победимова Светлана, Польшивная Елена, Поминова Елена, Пономарева Юлия, Посысаев Роман, Никитин Вячеслав, Пудова Наталья, Пушкарев Алексей, Размыслова Анна, Ратников Евгений, Ростовцев Евгений, Сасимович Андрей, Сенчук Светлана, Серебренникова Дарья, Силаев Валерий, Склюева Юлия, Суховерхова Ксения, Сушик Василий, Тарасов Евгений, Тарасов Юлиан, Татаркин Максим, Ткаченко Анна, Трунова Елена, Тулеуова Айнура, Уваровская Валентина, Урдиханова Алина, Хаустова Кристина, Хе Николай и Ющенко Максим, Чапкевич Михаил, Черникова Светлана, Чиковинский Дмитрий, Хрякова Ольга, Чевыкалов Дмитрий, Шамсутдинова Альбина, Шарипов Ильфир, Щемак Екатерина и Эрштейн Антон, а также ребята из г. Заозерный, из села Кемля, из поселка Клетня, из школы № 829 Москвы, из г. Инта и из села Подсосное.

У остальных участников конкурса есть все шансы стать лидерами в будущем (итоги конкурса будут определены после проведения всех туров).

Отметим ответ Валерия Силаева, разработавшего программу на языке Visual Basic, в которой моделируется состояние емкостей в результате выполнения тех или иных команд.

Конкурс № 43 для учащихся

Вы, конечно, знаете об азбуке Морзе — системе кодирования букв, использующей два символа: точку и тире. С ее помощью можно передавать текстовую информацию, например, по телеграфу или по радиации. Но как быть китайским телеграфистам и радистам, если китайские иероглифы — это не буквы, а знаки, обозначающие целые понятия: один знак — “дом”, другой — “дерево”, третий — “смотреть” и т.д. Предложите способ передачи текста,

изображенного в виде иероглифов, с помощью азбуки Морзе.

Ответы отправьте в редакцию до 1 марта по адресу: 121165, Москва, ул. Киевская, д. 24, “Первое сентября”, “Информатика” или по электронной почте: inf@1september.ru. Пожалуйста, четко укажите в ответе свои фамилию и имя, населенный пункт, номер и адрес школы, фамилию, имя и отчество учителя информатики.

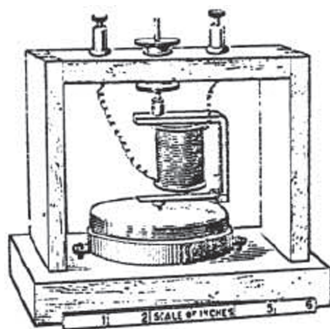
Учителю информатики: памятные даты февраля

Окончание. См. с. 1–2

14 февраля 1876 года Александр Грейам Белл (шотландец по происхождению, живший с 1871 г. в США) подал заявку на свое изобретение — «Телеграф, при помощи которого можно передавать человеческую речь» (телефон).

(Всего двумя часами позже заявку на «Устройство для передачи и приема вокальных звуков телеграфным способом» подал Э.Грей из Чикаго.)

Белл хорошо понимал значение и перспективы своего изобретения и уже в 1878 году так предсказывал его будущее [9]: «Можно представить себе, что кабели телефонных линий будут проложены под землей или подвешены в воздухе, их ответвления соединят жилые помещения, магазины, фабрики и т.д. с главным кабелем, идущим на центральную станцию, где все эти провода будут соединяться в нужных комбинациях, и, таким образом, установится непосредственная связь между двумя любыми точками в городе». И далее: «...в будущем телефонные линии соединят центральные станции разных городов, и человек, находящийся в одной части страны, сможет словесно общаться с человеком, находящимся совсем в другой ее части».



Телефонный аппарат
Белла

15 февраля 1930 года в городе Сенгилей Ульяновской области родился Евгений Андреевич Жоголев (1930–2003) — один из первых отечественных программистов и один из создателей трюичной «Сетуни», заслуженный профессор МГУ им. М.В. Ломоносова, полвека проработавший в этом учебном заведении.

15 февраля 1934 года в небольшом швейцарском городке Винтертуре (в

предместье Цюриха) родился Никлаус Вирт — один из самых выдающихся творцов языков программирования в XX столетии.

Никлаус Вирт известен прежде всего как создатель языка программирования Паскаль, ставшего прародителем нескольких языков программирования [10–12]. Помимо Паскаля, на счету Вирта такие великопленные разработки, как Модула-2, Oberon, а также многое другое [13].

15 февраля 1946 года в США состоялась демонстрация работы ENIAC (от *Electronic Numerical Integrator and Computer* — электронный цифровой интегратор и компьютер) — первого успешно функционировавшего электронного цифрового компьютера.

Использованные источники информации

1. *Частиков А.П.* Архитекторы компьютерного мира. СПб.: БХВ-Петербург, 2002.
2. *Токуров Т.* Не битом единым // *Hard'n'Soft* № 6/2003.
3. Чудо-машина // *Информатика* № 34/2003.
4. *Храмов Ю.А.* Физики. Биографический справочник. Изд. 2-е. М.: Наука, Гл. редакция физико-математической литературы, 1983.
5. Детская энциклопедия. Т. 5. М.: Издательство Академии педагогических наук РСФСР, 1960.
6. Материалы сайта www.computer-museum.ru.
7. *Мусский С.А.* 100 великих чудес техники. М.: Вече, 2005.
8. *Чолаков В.* Нобелевские премии. Ученые и открытия: Пер. с болг. М.: Мир, 1986.
9. *Шарле Д.Л.* По всему земному шару: Прошлое, настоящее и будущее кабелей связи. М.: Радио и связь, 1985.
10. Язык компьютера: Пер. с англ. М.: Мир, 1989.
11. *Малыхина М.П., Частиков А.П.* Языки программирования: Паскаль // *Новое в жизни, науке, технике. Сер. "Вычислительная техника и ее применение"* № 6/1990.
12. *Пройдаков Э.М., Теплицкий Л.А.* Англо-русский толковый словарь по вычислительной технике, Интернету и программированию. Изд. 2-е, испр. и доп. М.: Издательско-торговый дом «Русская Редакция», 2000.
13. *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.
14. «Наука и жизнь» № 11/1981, с. 8.

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. Семина,
индекс подписки — 32033;

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Гл. редактор
С.Л. Островский
Зам. гл. редактора
А.И. Сенокосов
Редакция
Е.В. Андреева
Д.М. Златопольский (редактор
вкладки «В мир информатики»)
Л.Н. Картвелишвили
С.Б. Кишкина
Н.П. Медведева
Ю.А. Первин (редактор вкладки
«Началка»)
Корректор
Е.Л. Володина
Дизайн и верстка
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 6500 экз.
Срок подписания в печать по графику 19.01.2006.
Номер подписан 19.01.2006.
Заказ № 615503
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>