

# ИНФОРМАТИКА

## «Эксель!»



издательский дом  
**Первое сентября**

| ИНФОРМАТИКА | Индексы подписки | Почта России – 79006 (инд.); – 79574 (орг.) | Роспечать – 32291 (инд.); – 32591 (орг.)



# В НОМЕРЕ

**3** **Новость № 1**  
О проведении Всероссийского съезда учителей информатики в МГУ им. М.В. Ломоносова

**4** **ЕГЭ**  
Освой КуМир за 6 часов: последние четыре часа занятий

**9** **Тема номера**  
Ёксель!

**18** **Алгоритмизация и программирование**  
Дело в шляпе

**23** **Газета для пытливых учеников и их талантливых учителей**  
"В МИР ИНФОРМАТИКИ" № 156

# На диске



На диске к этому номеру размещены:  
— лабораторные работы по теме "Электронные таблицы" к статье "Ёксель!";  
— дистрибутив системы КуМир и шаблоны задач к статье "Освой КуМир за 6 часов"; а также размещен PDF-файл с первой частью указанной статьи, которая была опубликована в последнем номере прошлого года;  
— коды программ для построения поверхностей к статье "Дело в шляпе".

# ИНФОРМАТИКА

Учебно-методическая газета для учителей информатики  
Основана в 1995 г.  
Выходит два раза в месяц

**РЕДАКЦИЯ:**  
гл. редактор С.Л. ОСТРОВСКИЙ  
редакторы

Е.В. АНДРЕЕВА,  
Д.М. ЗЛАТОПОЛЬСКИЙ  
(редактор вкладки  
"В мир информатики")

верстка Н.И. ПРОНСКАЯ  
корректор Е.Л. ВОЛОДИНА  
секретарь Н.П. МЕДВЕДЕВА

Фото: фотобанк Shutterstock  
Газета распространяется по подписке  
Цена свободная  
Тираж 3000 экз.  
Тел. редакции: (499) 249-48-96  
E-mail: inf@1september.ru  
http://inf.1september.ru

ИЗДАТЕЛЬСКИЙ ДОМ  
"ПЕРВОЕ СЕНТЯБРЯ"

**Главный редактор:**  
Артем Соловейчик  
(Генеральный директор)

**Коммерческая деятельность:**  
Константин Шмарковский  
(Финансовый директор)

**Развитие, IT и координация проектов:**  
Сергей Островский  
(Исполнительный директор)

**Реклама и продвижение:**  
Марк Сартан

**Мультимедиа, конференции и техническое обеспечение:**  
Павел Кузнецов

**Производство:**  
Станислав Савельев

**Административно-хозяйственное обеспечение:**  
Андрей Ушков

**Дизайн:**  
Иван Лукьянов, Андрей Балдин

**Педагогический университет:**  
Валерия Арсланьян (ректор)

ГАЗЕТЫ  
ИЗДАТЕЛЬСКОГО ДОМА:

**Первое сентября** – Е.Бирюкова  
**Английский язык** – А.Громушкина  
**Библиотека в школе** – О.Громова

**Биология** – Н.Иванова  
**География** – О.Коротова

**Дошкольное образование** – М.Аромштам  
**Здоровье детей** – Н.Сёмина

**Информатика** – С.Островский  
**Искусство** – М.Сартан

**История** – А.Савельев  
**Классное руководство и воспитание**

**школьников** – О.Леонтьева  
**Литература** – С.Волков

**Математика** – Л.Рослова  
**Начальная школа** – М.Соловейчик

**Немецкий язык** – М.Бузоева  
**Русский язык** – Л.Гончар

**Спорт в школе** – О.Леонтьева  
**Управление школой** – Я.Сартан

**Физика** – Н.Козлова  
**Французский язык** – Г.Чесновицкая

**Химия** – О.Блохина  
**Школьный психолог** – И.Вачков

УЧРЕДИТЕЛЬ:  
ООО "ЧИСТЫЕ ПРУДЫ"

**Зарегистрировано**  
**ПИ № 77-72230**  
**от 12.04.2001**

в Министерстве РФ по делам печати  
Подписано в печать: по графику 16.12.2010, фактически 16.12.2010  
Заказ №

Отпечатано в ОАО "Чеховский полиграфический комбинат"  
ул. Полиграфистов, д. 1, Московская область, г. Чехов, 142300

**АДРЕС ИЗДАТЕЛЯ:**  
ул. Киевская, д. 24, Москва, 121165  
**Тел./факс:** (499) 249-31-38

**Отдел рекламы:**  
(499) 249-98-70  
<http://1september.ru>

**ИЗДАТЕЛЬСКАЯ ПОДПИСКА:**  
**Телефон:** (499) 249-47-58  
**E-mail:** [podpiska@1september.ru](mailto:podpiska@1september.ru)



Документооборот  
Издательского дома  
"Первое сентября" защищен  
антивирусной программой  
Dr.Web



## О проведении Всероссийского съезда учителей информатики в МГУ им. М.В. Ломоносова 24–26 марта 2011 года

### Первое информационное сообщение

Московский государственный университет им. М.В. Ломоносова возрождает традиции проведения Всероссийских съездов учителей-предметников. В октябре 2010 года в стенах МГУ состоялся съезд учителей математики, летом 2011 года пройдут съезды школьных учителей физики и биологии.

**24–26 марта 2011 года** в МГУ состоится **Всероссийский съезд учителей информатики**, на котором пройдет широкое обсуждение состояния и перспектив развития школьного образования в области информатики и информационно-коммуникационных технологий в контексте основных принципов Национальной образовательной инициативы “Наша новая школа”.

На съезд приглашаются:

- учителя информатики и ИКТ общеобразовательных школ;
- специалисты по педагогике и методике преподавания информатики и ИКТ;
- руководители образовательных учреждений;
- представители органов управления образованием;
- учителя математики, физики и других дисциплин;
- преподаватели высших учебных заведений.

Работа съезда будет организована в виде тематических секций и “круглых столов”, посвященных различным аспектам современного образования в области информатики и ИКТ.

Для участия в работе съезда **с докладом** необходимо в период с 20 декабря 2010 года по 15 февраля 2011 года подать заявку с тезисами в Программный комитет съезда. Требования к тезисам размещены на сайте съезда. Решение об участии в работе съезда с докладом на одной из секций принимает Программный комитет съезда по результатам рассмотрения поданной заявки.

Для участия в работе съезда **без доклада** необходимо в период с 1 января по 15 февраля 2011 года подать заявку в Организационный комитет через официальный сайт съезда. Регистрация индивидуальная.

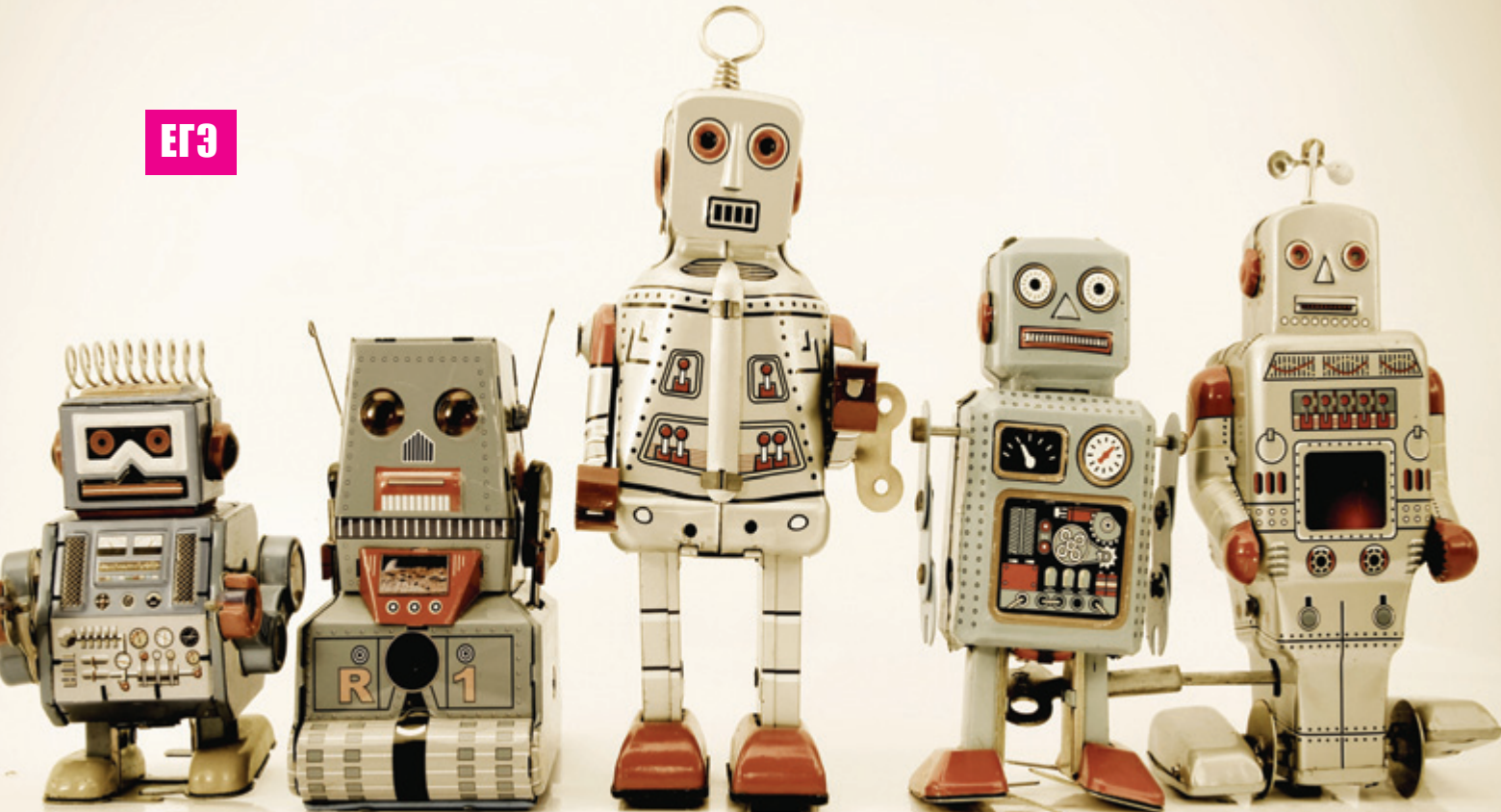
Приглашения для участия в работе съезда будут разосланы до 1 марта 2011 года.

Участие в съезде без организационного взноса.

Организационный комитет не оплачивает проезд и проживание участников съезда и информирует о том, что количество участников, которым могут быть предоставлены места для проживания во время съезда, строго лимитировано. В приглашении будет указано, предоставляется ли участнику место в гостинице МГУ.

*Формы заявок и вся дополнительная информация о проведении съезда размещается на официальном web-сайте съезда по адресу [it.teacher.msu.ru](http://it.teacher.msu.ru).*





## Освой КуМир за 6 часов:

последние четыре часа занятий

**А.Г. Леонов,  
Москва**

*Окончание.  
Первая часть  
статьи была  
опубликована  
в № 24/2010.  
Для наших новых  
подписчиков  
мы размещаем  
электронный  
вариант первой  
части статьи  
на прилагаемом  
компакт-диске.*

Напомним, что в кодификаторе элементов содержания и требований к уровню подготовки выпускников общеобразовательных учреждений для проведения в 2011 году единого государственного экзамена по “Информатике и ИКТ” приведен список из 15 шаблонов возможных алгоритмических задач для подраздела 1.1 перечня требований к уровню подготовки выпускников, достижение которого проверяется на едином государственном экзамене по “Информатике и ИКТ”<sup>1</sup>. В первой части статьи мы рассмотрели решение четырех первых задач из списка, рассмотрим здесь еще несколько интересных, с нашей точки зрения, задач:

5. Нахождение сумм, произведений элементов данной конечной числовой последовательности (или массива).

6. Нахождение минимального (максимального) значения в данном массиве и количества элементов, равных ему, за однократный просмотр массива.

7. Нахождение второго по величине (второго максимального или второ-

го минимального) значения в данном массиве за однократный просмотр массива.

8. Операции с элементами массива, отобранными по некоторому условию (например, нахождение минимального четного элемента в массиве, нахождение количества и суммы всех четных элементов в массиве).

9. Обработка отдельных символов данной строки. Подсчет частоты появления символа в строке.

10. Работа с подстроками данной строки с разбиением на слова по пробельным символам. Поиск подстроки внутри данной строки, замена найденной подстроки на другую строку.

**Задача № 5.** Уточним ее формулировку. В файле “5in.txt” находится последовательность целых чисел. Нужно найти сумму элементов этой последовательности и записать результат в файл “5out.txt”.

С сайта <http://test.kumir.su> со страницы с этой задачей загружаем шаблон алгоритма

**алг** сумма последовательности

<sup>1</sup> “Единый государственный экзамен по “Информатике и ИКТ”. Кодификатор элементов содержания и требований к уровню подготовки выпускников общеобразовательных учреждений для проведения в 2011 году единого государственного экзамена по “Информатике и ИКТ”. Подготовлен Федеральным государственным научным учреждением “Федеральный институт педагогических измерений”.







Обратим внимание, что заготовка алгоритма практически полностью совпадает (с точностью до названия и имен файлов) с задачей № 6:

**использовать файлы**

**целтаб** M[1:100]

**алг** Нахождение 2-го максимального  
*. дано* | Элементы последовательности  
           | (целые неотрицательные числа)  
*/* находятся в файле 7in.txt  
*/* Число элементов в последовательности  
*/* (в том числе пустой) не более 100  
*. надо* | Найти 2-й максимальный по величине  
           | элемент последовательности  
*/* и вывести в файл 7out.txt или -1,  
*/* если такого нет

**нач** цел Ключ, n  
*. открыть* на чтение ("7in.txt", Ключ)  
*. n := 0*  
*. нц пока не* конец файла(Ключ)  
*. . n := n + 1*  
*. . ф\_ввод* Ключ, M[n]  
*. кц*  
*. закрыть* (Ключ)  
*.*  
*. если не* существует файл("7out.txt")  
*. . то* создать файл("7out.txt")  
*. все*  
*. открыть* на запись("7out.txt", Ключ)  
*. ф\_вывод* Ключ, Второй максимум(n)  
*. закрыть* (Ключ)  
*.*

**кон**

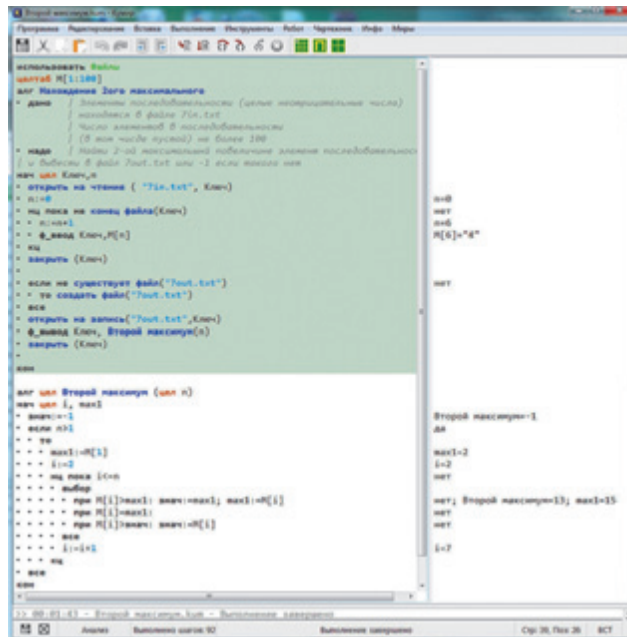
Что касается функции “Второй максимум”, то попытаемся ее вычислить так же, как и функцию “Число минимальных”. Для этого составим таблицу, предполагая, что для нахождения второго максимума нам потребуется знать первый максимум:

Число элементов	max	max2
0	Нет	-1
1	M[1]	-1
i > 1	M[i], при M[i] > max	max, при M[i] > max ничего не делать, при M[i] = max M[i], при M[i] > max2

В приведенной таблице есть только одна тонкость при ее программировании: при M[i] > max сначала значение max копируется в max2, а уже потом max устанавливается равным M[i]. Теперь понятно, как записать алгоритм вычисления функции на алгоритмическом языке (см. скриншот).

**Задача № 8.** Частный случай. Нахождение минимального четного элемента в массиве.

Прежде всего отметим, что заготовка алгоритма, в которой мы читаем массив из файла и выводим результат в файл, полностью аналогична двум предыдущим задачам. Поэтому надеемся, что читатель самостоятельно разберется с ней, загрузив с сайта <http://test.kumir.su>.



В принципе алгоритм задачи похож на поиск минимума, однако из массива сначала осуществляется некоторая выборка — четные элементы массива:

*. . . . если* mod(M[i], 2) = 0

— остаток от деления которых на 2 равен нулю.

Обратите внимание, что четного минимума вообще может не быть, в случае, если все элементы массива нечетные. В нашем примере мы предполагаем, что все элементы массива неотрицательные числа. Конечно, мы сузили область значений элементов массива, дабы упростить алгоритм. Но это легко исправить. Как? Оставляем задачу найти ответ на этот вопрос вдумчивому читателю.

Как и в предыдущей задаче, попытаемся найти взаимосвязь функции с ее предыдущим значением:

Число элементов	Четный min
0	-1
i > 0	При M[i] четный: min = M[i], при min = -1 min = M[i], при min > M[i]

В таблице изменения Min произойдут, если только M[i] — четный.

Таким образом, запишем алгоритм “Четный минимальный”:

**алг** цел Четный минимальный (цел n)  
**нач** цел i  
*. знач := -1*  
*. если* n > 0  
*. . то*  
*. . . нц для i от 1 до n*  
*. . . . если* mod(M[i], 2) = 0  
*. . . . . то*  
*. . . . . выбор*  
*. . . . . . при знач = -1 : знач := M[i]*  
*. . . . . . при знач > M[i] : знач := M[i]*  
*. . . . . все*  
*. . . . все*  
*. . . кц*  
*. все*  
**кон**

В конце задачи приведем шаблон заготовки на задаче с массивами:

**использовать файлы**

**целтаб** M[1:100]

**алг**

. дано

. надо

**нач** цел Ключ, n

. открыть на чтение ("in.txt", Ключ)

. n := 0

. **нц** пока не конец файла(Ключ)

. . n := n + 1

. . **ф\_ввод** Ключ, M[n]

. **кц**

. закрыть (Ключ)

.

. **если** не существует файл("out.txt")

. . **то** создать файл("out.txt")

. **все**

. открыть на запись("out.txt", Ключ)

. **ф\_вывод** Ключ, F (n)

. закрыть (Ключ)

.

**кон**

**алг** цел F (цел n)

**нач**

**кон**

**Совет № 20.** Кроме использования исполнителя Файлы, есть еще способ общения с внешним миром — команды **ввод** и **вывод**.

**нс** (новая строка) и " " в команде **вывод** позволяет оживить оформление.

**Совет № 21.** Операции со строками (**лит**) немного отличаются от операций с целыми величинами. Так, **+** означает склейку двух строк, **длин(строка)** возвращает длину строки в символах, **строка[x:y]** — вырезку из строки от символа с номером x до символа с номером y, а **строка[x]** — символ на позиции x.

**Задача № 9.** Подсчет частоты появления символа в строке.

Алгоритм подсчета частоты символа в строке имеет два аргумента: символ и строку. Результат — сколько раз символ встречается в строке. Для этого можно использовать цикл, в нем просматриваем строку посимвольно и сравниваем с эталонным символом. Если обнаружено совпадение, то счетчик символов надо увеличить на 1:

**алг** цел Число Символов(арг сим x, лит стр)

. **надо** / **знач** = число вхождения символа x / в строку стр

**нач** цел инд

. **знач** := 0

. **нц** для инд от 1 до длин(стр)

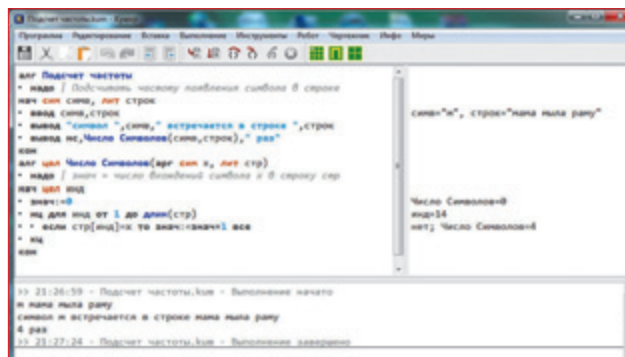
. . **если** стр[инд] = x **то** **знач** := **знач** + 1

. . **все**

. **кц**

**кон**

Работа общего алгоритма выглядит так:



**Задача № 10.** Рассмотрим лишь первую часть задачи, слегка модифицировав ее: подсчитать число слов в строке.

Строка — последовательность символов: пробелов и непробелов. Слово — это последовательность одного или нескольких непробелов, ограниченная пробелами с обеих сторон или стоящая в начале или конце строки.

Будем подсчитывать слова, когда прочитаем первую букву нового слова. Это означает, что предыдущий символ был пробел, а текущий — непробел. Это условие подходит для всех случаев, кроме начала строки. Тогда можно считать, что предыдущий символ перед началом строки — пробел.

У нас получится следующий алгоритм:

**алг** цел Число Слов(арг лит строка)

. **дано** / Последовательность символов, / состоящая из пробелов и непробелов

. **надо** / Ответ = число слов / в последовательности

**нач** цел инд, сим последний, а

. **знач** := 0; последний := " "

/ строка начинается пробелом

. **нц** для инд от 1 до длин(строка)

. . а := строка[инд]

. . **если** последний = " " и а <> " "

. . . **то** **знач** := **знач** + 1

. . . / слово опознается по его началу

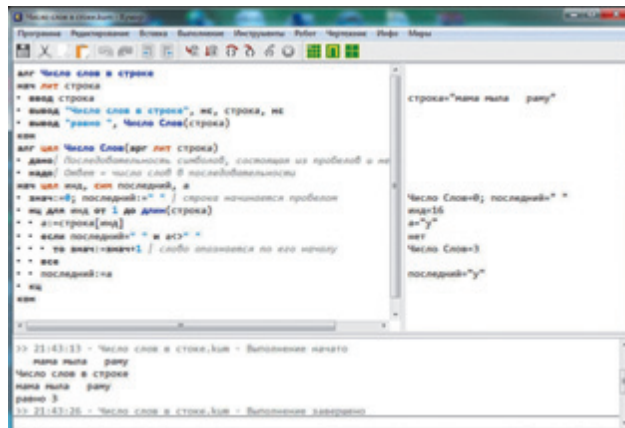
. . **все**

. . последний := а

. **кц**

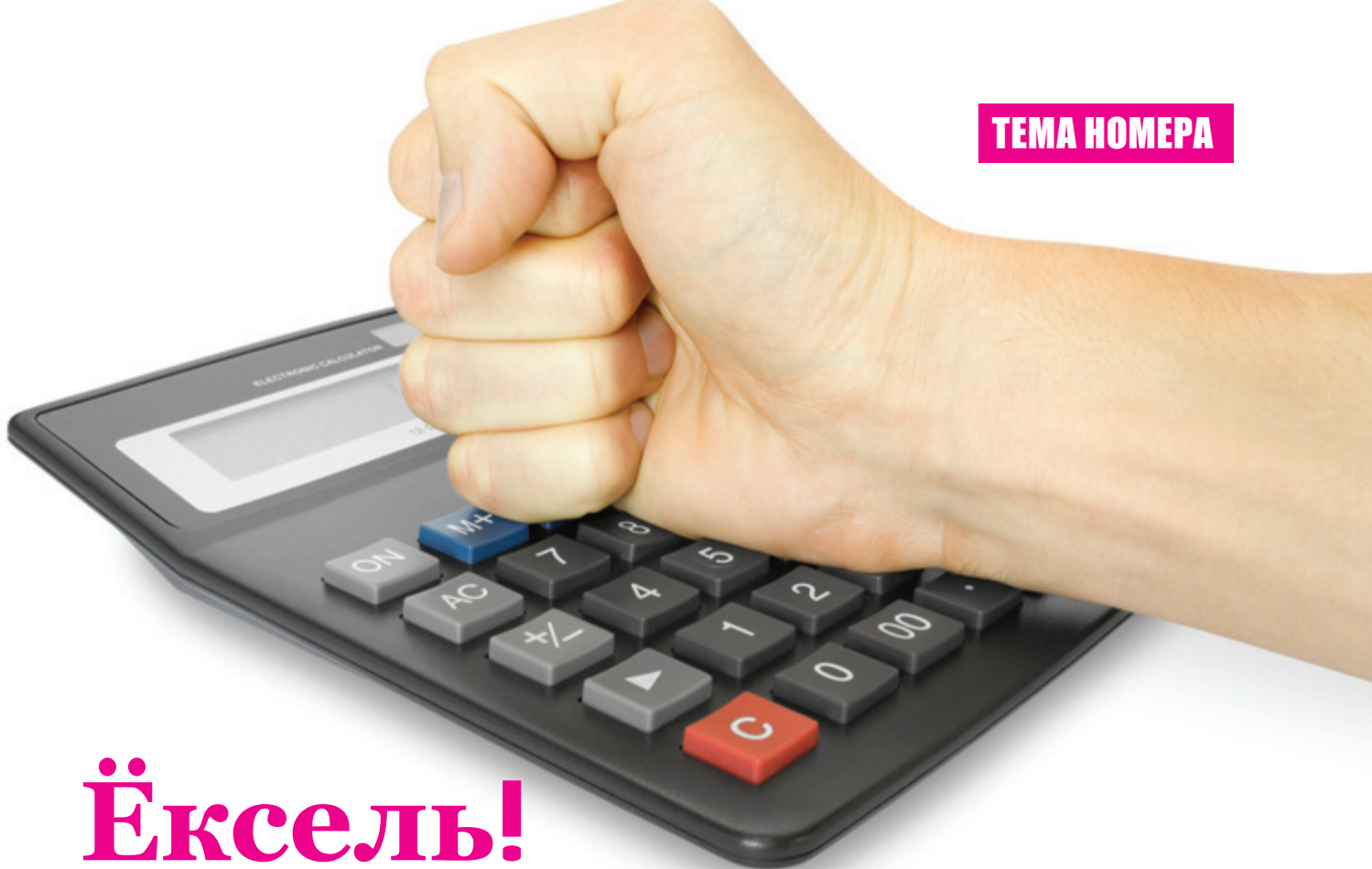
**кон**

Проверим работу алгоритма:



В заключение хочется порекомендовать читателю загрузить, проверить, решить и подумать над другими примерами, задачами и заготовками программ с сайта <http://test.kumir.su>.





# Ёксель!

**А.И. Сенокосов,  
г. Екатеринбург**

Пару лет назад, работая над созданием ознакомительного курса информационных технологий для 6-го класса, я столкнулся с необходимостью разработки нескольких лабораторных работ по теме “Электронные таблицы”. Почему-то мне казалось, что проблема решается исключительно быстро: достаточно прошерстить Интернет, и уж там-то, да по такой расхожей теме точно найдется что-нибудь подходящее.

Действительно, на запрос “Лабораторная Excel” выдается больше сотни тысяч ссылок. Каково же было мое удивление, когда я не смог найти среди них ни одной, которая бы соответствовала моему представлению о нормальной, интересной и познавательной лабораторной работе. Особенно для 6-го класса.

Лучшее, что я нашел, повергает в уныние:

1. Запустите программу Excel.
2. Наберите свое имя в ячейке A1 и зафиксируйте текст.
3. Переместите содержание этой ячейки в ячейку B3 простейшим способом: наведите указатель мыши на рамку вашей ячейки, и когда указатель примет вид черного тонкого крестика, перетяните его в ячейку B3, удерживая левую клавишу мыши.

4. Теперь скопируйте содержание этой ячейки в ячейку D4 простейшим способом: наведите указатель мыши на рамку вашей ячейки, и когда указатель примет вид черного тонкого крестика, перетяните его в ячейку B3, удерживая левую клавишу мыши и клавишу **Ctrl**.

5. Удалите содержание всех ячеек, в которых содержится ваше имя, разными способами. Для этого:

- выделите ячейку A1 и нажмите клавишу **Delete**;
- выделите столбец B3 и выполните команду **Правка > Очистить > Все**;
- выделите всю таблицу и нажмите клавишу **Delete**.

Не вдаваясь в методические тонкости, в которые я, возможно, просто не умею вдаваться, отметим, что подобные лабораторные работы ввергают в уныние не только меня, но и подавляющее большинство учеников. Они начинают воспринимать информационные технологии как некий оторванный от жизни и весьма скучный предмет, в котором самое главное — запомнить функциональные клавиши и нужные кнопочки. Ну и, разумеется, они совершенно не подходят для начального обзорного курса 6-го класса.

С другой стороны, у меня имелся вводный курс, разработанный в свое вре-

мя для 7-го класса (более десяти лет назад он был издан в Екатеринбурге: *Сенокосов А.И., Гейн А.Г.* Информатика, 7-й класс. Екатеринбург: ИД “Сократ”, 1999), но он уже несколько устарел, что для нашего предмета неудивительно. В очередной раз подтвердилась суровая истина: “Если сам не напишешь нормальный учебник — не напишет никто” ©. Пришлось взяться за дело и на основе старого курса разработать несколько новых лабораторных работ. Ниже как раз и приведено то, что получилось в итоге. Добавляю лишь, что курс опробован в течение двух лет и, как мне кажется, показал свою эффективность. Безусловно, вовсе необязательно выполнять все лабораторные работы. Возможно, кому-то некоторые из них покажутся излишне трудными, другие просто не влезут в сетку часов. Но среди них, во всяком случае, нет ни одной скучной. Такой, где функциональные возможности MS Excel изучались бы сами по себе, в отрыве от решаемой задачи.

## Как я рассказываю детям о том, что такое электронные таблицы

Наверное, вы уже знаете, чем компьютер принципиально отличается от простого калькулятора. Одно дело — сидеть с калькулятором и вычислять, например, заработную плату каждого из работников, и совсем другое — перемножать, складывать и вычитать целыми столбцами. Или просто вставить в ячейку таблицы необходимую для расчетов формулу и сразу получить нужный результат.

Иными словами, компьютер может автоматически пересчитывать все данные в таблице при изменении некоторых из них. Бухгалтеру, например, частенько требуется пересчитать весь кварталный отчет (а это несколько десятков показателей), если возникает необходимость заменить некоторые исходные числа. Или заново начислить заработную плату всем работникам при изменении всего лишь одного числа — минимальной заработной платы.

Подобные проблемы возникают не только у бухгалтеров. Диспетчерам при составлении графиков движения транспорта, экспериментаторам при проведении серий опытов и многим другим приходится решать задачи, в которых изменение значения какого-то одного параметра требует пересчета большого числа результатов.

И компьютер способен учесть огромный объем данных и сослужить добрую службу, неоднократно проводя однообразные вычисления. Для этого существуют специальные программы — **электронные таблицы**.

Но прежде чем знакомиться с ними, давайте подумаем, а что требуется для расчета все той же заработной платы.

Ну, во-первых, фамилии и инициалы сотрудников, а также ставки, количество отработанных ими часов, льготы по налогообложению и т.п. Эту информацию надо просто занести в компьютер для использования в дальнейших вычислениях. Такого сорта информация называется *исходными данными*.

Во-вторых, необходимо знать формулы, по которым рассчитываются заработная плата, налоги, премияль-

ные и т.п. Получающиеся при этом числа называются *рассчитываемыми данными*, или *результатами*.

Электронная таблица позволяет хранить в табличной форме большое количество исходных и рассчитываемых данных, а также связи между ними (то есть формулы). Но главное — при изменении исходных данных все результаты автоматически пересчитываются и изменяются прямо у вас на глазах.

Электронную таблицу можно себе представить, как здание с подвалом: на видимом этаже — привычные для нас числа, буквы и другие символы — то есть исходные и рассчитываемые данные, записанные в ячейки обычной таблицы, а в подвале — невидимые с первого взгляда формулы, позволяющие получить результаты.

Ячейки таблицы, у которых в верхнем слое — число, а в нижнем ничего нет, — это, как легко понять, ячейки с исходными данными. Они, напомним, заносятся вручную и компьютером никак не изменяются.

Ячейка таблицы с формулой в нижнем слое — это ячейка, в которую компьютер самостоятельно записывает результат. Вычислить его — забота компьютера.

Идея проста, однако додумались до нее спустя целых двадцать лет после того, как стали применять компьютеры в бухгалтерии. Сделал это Дэниел Бриклин в 1979 году. Вместе с программистом Робертом Фрэнкстоном они и создали первую в мире электронную таблицу с названием “Визикалк”, что по замыслу создателей означало “Визуальный калькулятор”.

Даже одна эта программа оправдывала в глазах пользователей приобретение персонального компьютера: более чем 100 тысяч человек купили персональный компьютер ради возможности работать с “Визикалком”. Экономисты, захлебываясь от восторга, писали, что с помощью этой программы можно почти мгновенно определить прибыль компании, если заработная плата, например, вырастет на 6% при одновременном увеличении производительности труда на 3,5% и уменьшении цены на готовую продукцию на 7% с ожидаемым увеличением сбыта на 14%.

Посмотрим внимательно на электронную таблицу.

	Всего	Есть	Есть	На 4 и 5	Отличников	Процент
	учеников	двойки	тройки			успеваемости
5-е классы	89	7	55	23	4	92,13%
6-е классы	91	3	49	35	4	96,70%
7-е классы	105	3	56	41	5	97,14%
8-е классы	93	3	64	19	7	96,77%
9-е классы	82	5	47	29	1	93,90%
10-е классы	50	0	24	23	3	100,00%
11-е классы	46	1	16	25	4	97,83%
По школе	556	22	311	195	28	96,04%

Внешний вид электронной таблицы на экране компьютера

Несмотря на сложную двухуровневую структуру, видимый этаж электронной таблицы почти ничем не отличается от самого обычного документа, созданного в текстовом редакторе.



Конечно, страничка разбита на *столбцы*, обозначенные буквами латинского алфавита, и на *строки*, пронумерованные целыми числами, но само это разбиение при выводе на принтер, как правило, делается невидимым, и по внешнему виду документа невозможно понять, сделан он с помощью текстового редактора или электронной таблицы.

Благодаря разбиению на пронумерованные строки и поименованные столбцы каждая ячейка электронной таблицы имеет свой собственный **адрес**.

Так, например, в ячейке **B1** написан заголовок таблицы. Вообще-то столбец **B** не такой уж и широкий, чтобы в него влез весь заголовок целиком, но таблица устроена так, что если в правой соседней ячейке ничего нет, то текст перекрывает и ее. Если же в ячейку **C1** что-нибудь поместить, то вместо слов «Итоги первой четверти по школе» мы бы увидели только «Итоги п».

Ширину столбцов можно при необходимости менять, а некоторые строки — пропускать. Это позволяет красиво оформить документ.

К сожалению, не во всех электронных таблицах есть возможность увидеть подвальный этаж целиком, но всегда виден подвал той ячейки, на которой стоит курсор в виде прямоугольника. Как вы думаете, на какой ячейке он стоит на нашем рисунке?

Содержимое подвала отображается в специально отведенном для этого месте (на иллюстрации на с. 10 оно изображено в рамочке над таблицей, выделенной желтым маркером). Что же касается формулы, то, внимательно посмотрев на таблицу, вы, наверное, догадаетесь, что за формулы стоят в ячейках, отмеченных красными числами.

Для электронной таблицы существуют специальные правила ее заполнения. Дело в том, что данные в ячейках могут быть трех типов: числа, текст и формулы. Электронная таблица по-разному их обрабатывает. Так, текст она может продлить вправо, даже если он выходит за рамки предназначенной для него ячейки, а формулы вообще помещает в подвальный невидимый этаж.

Правила заполнения такие:

- Если среди символов, вводимых в ячейку, есть буквы или нечто, чего не может быть в числе, то это — текст. Например: «Василий», или «10.234.245», или «a1234».
- Если вводится правильная запись числа — то это число. Например: 43,23 или 234 (разумеется, без кавы-

чек; первый символ « (кавычки) сразу же указывает, что тип данных — Текст).

- Если ввод начинается со знака «=» или другого специально оговоренного знака, то это — формула, располагаемая в подвальном этаже таблицы.

Электронная таблица имеет, помимо всего прочего, целый набор стандартных функций, облегчающих жизнь пользователю. Как вы думаете, какая формула стоит в ячейке **B14**? Нет, разумеется, можно написать такое:

$$=B6+B7+B8+B9+B10+B11+B12$$

и это будет правильно. Ну а если надо будет сложить не семь, а двадцать семь или сто семь чисел? Для облегчения работы с таблицей имеется стандартная функция — *суммирование* содержимого *блока ячеек*:

$$=СУММ(B6:B12)$$

Это и проще, и понятнее.

У электронной таблицы есть много разных операций над *блоками ячеек*. Это — поиск максимального или минимального элемента, расчет среднего значения и т.д. **Блок ячеек** можно скопировать или перенести в другое место таблицы.

Надо только помнить, что *блок ячеек* в электронной таблице — это все ячейки, заполняющие некоторый прямоугольник (в примере на иллюстрации на с. 10 — прямоугольник размером 1 × 7). Для того чтобы электронная таблица знала, какой блок ей надо обрабатывать, указывают через двоеточие *адреса* ячеек, стоящих в левом верхнем и правом нижнем углах прямоугольника (например, **C6:E12**).

Одним словом, современная электронная таблица максимально приспособлена для быстрой и удобной работы, и пользующийся ею один человек вполне способен заменить целое бухгалтерское бюро прошлого.

	A	B	C	D	E	F
1	Река	Длина реки, км	Число городов	Население городов, тыс. чел	На 1 город приходится в среднем (км длины реки)	Число жителей городов на 1 км длины реки
2	Волга	3 531	69	11 870,40		
3	Ока	1 500	22	3 503,30		
4	Клязьма	686	21	12 153,90		
5	Дон	1 870	20	2 753,90		
6	Кама	1 805	18	2 496,80		
7	Обь	3 650	14	3 080,80		
8	Енисей	3 487	13	1 591,80		
9	Кубань	870	11	1 464,80		
10	Москва	473	10	11 004,30		
11	Белая	1 430	10	1 784,70		
12	Вятка	1 314	10	697,6		
13	Томь	827	7	1 911,60		
14	Ангара	1 779	7	1 305,00		
15	Исеть	606	6	1 612,90		
16	Северная Двина	744	6	692,3		
17	Тура	1 030	6	652,5		
18	Десна*	415	6	504,2		
19	Терек	623	6	487,9		
20	Лена	4 400	6	340,6		
21	Хопёр	979	6	245,7		
22	Пьяма	603	6	215,3		
23	Протва	282	6	194,1		
24	Урал*	1 346	5	1 318,00		
25	Чусовая	592	5	1 300,90		
26	Амур	2 824	5	1 146,80		
27	Сура	841	5	608,5		
28	Болва	208	5	524,8		
29	Сунжа	278	5	414,3		
30	Кума	802	5	278,3		
31						

### ДЛЯ УЧИТЕЛЕЙ МАТЕМАТИКИ, ИНФОРМАТИКИ, ИСТОРИИ И УЧЕНИКОВ



М.А.Цайгер.

«Арифметика в Московском государстве XVI века»,  
Безр-Шева, Берилл, 2010.

**В** книге рассмотрены темы: славянская система записи чисел кириллическими буквами, системы сокращённой записи чисел в решётках, стандарты разметки расчётных столов для счёта костями. Даны примеры сложения костями целых неименованных и именованных чисел (денежных сумм), третних и четвертных и иных видов дробей, преобразования третних дробей в четвертные и наоборот.



Цена книги с пересылкой по России 260 руб.  
Адрес для дополнительной информации  
и заказа: [m\\_tsayger@hotmail.com](mailto:m_tsayger@hotmail.com)



## Лабораторная работа № 1. Реки России

Таблица “Реки России” представлена на с. 11.

В предложенной таблице сделайте следующее:

1. Запишите подходящие формулы в ячейки E2 и F2. При этом постарайтесь воспользоваться тем, что имена ячеек в формуле появляются при щелчке мышкой по указанной ячейке.
2. Продлите формулы до конца таблицы. Для этого переведите указатель мыши в правый нижний угол ячейки так, чтобы он изменился, превратившись в черный крестик. Нажмите левую клавишу мыши и продлите формулу вниз на все ячейки.
3. Отсортируйте таблицу таким образом, чтобы показать наиболее экологически загруженные реки. То есть те, на 1 км длины которых приходится больше всего жителей городов.
4. Оформите таблицу:
  - Дав ей заголовок “Реки России”. Для этого надо добавить новую строку.
  - Выделите другим цветом столбец с названием рек и заголовки столбцов.
  - Выделите маркером реку (реки), протекающую в вашей местности.

## Лабораторная работа № 2. Обмен валюты

Предположим, в банк пришел клиент, который захотел поменять рубли на одну из иностранных валют. Перед кассиром на экране компьютера находится таблица обмена валют.

Как вы уже догадались, в ячейки столбца С, отмеченные розовой заливкой, требуется вставить формулы. Думаем, вы без проблем вставите формулу в ячейку С3.

Тут нас подстерегает первая неожиданность. Оказывается, в таблице, скопированной из Интернета, вместо привычных нам запятых в числах стоят точки. Исправьте это, воспользовавшись закладкой главного меню “Найти и выделить”. Замените все точки на запятые, как это принято в России.

Следующая проблема состоит в том, что просто так продлить формулу не удастся. Можете попробовать, что из этого выйдет, и затем отменить операцию с помощью отката назад. Для того чтобы не вводить фактически одну и ту же формулу, воспользуемся возможностью дать ячейке уникальное имя.

- Щелкните правой клавишей мышки на ячейке, содержащей количество меняемых рублей;
- Выберите в выпадающем меню пункт “Имя диапазона”;
- Дайте любое имя ячейке.

После этого введите в ячейку С3 формулу еще раз. Если вы вводите ее, указывая мышкой на ячейку с количеством обмениваемых рублей, то вместо С3 в формуле появится данное вами имя.

Вот теперь формулу можно продлевать.

1	Валюта	Курс	1500 рублей
2	Курс доллара США	30.8604	
3	Курс евро	40.9394	
4	1 Австралийский доллар	30.4253	
5	1000 Белорусских рублей	204081,0000	
6	10 Датских крон	54.8113	
7	100 Казахских тенге	20.9415	
8	1 Канадский доллар	30.57	
9	10 Китайских юаней Жэньминьби	46.3724	
10	1 Новая турецкая лира	20.5162	
11	10 Норвежских крон	51.2648	
12	1 СДР	47.359	
13	1 Сингапурский доллар	23.5576	
14	10 Украинских гривен	38.7426	
15	1 Фунт стерлингов Соединенного Королевства	48.7687	
16	10 Шведских крон	44.8324	
17	1 Швейцарский франк	31.4581	
18	100 Японских йен	36.857	
19			
20			

Таблица обмена валют

## Лабораторная работа № 3. Экскурсионная поездка

В конце учебного года администрация школы организовала для всех желающих экскурсию в близлежащий заповедник и, получив информацию из классов о числе экскурсантов, заказала соответствующее количество автобусов.

Таблица 1

Класс	Едут на экскурсию	Первый автобус	Второй автобус	Третий автобус	Четвертый автобус
6-й "А"	23	23			
6-й "Б"	17	17			
6-й "В"	14	5	9		
6-й "Г"	7		7		
7-й "А"	18				
7-й "Б"	5				
7-й "В"	19				
7-й "Г"	22				
8-й "А"	21				
8-й "Б"	11				
8-й "В"	15				



Зная, что в каждый автобус входит ровно 45 пассажиров, завуч по внеклассной работе уже начала было заполнять таблицу (см. табл. 1), но тут возмутилась классный руководитель 6-го “В” класса. Конечно, всем ребятам из одного класса хотелось бы ехать в одном автобусе. Завуч, понятное дело, заявила в ответ, что все равно так не получится и кому-то придется пересесть в другой автобус. Классный руководитель, в свою очередь, не пожелала быть этим кем-то...

Так бы и разгорелся конфликт, но тут мимо пробежали компьютерно-грамотные шестиклассники и взялись с помощью электронной таблицы рассадить все классы таким образом, чтобы все ребята из одного класса ехали в одном автобусе и ни один автобус не был бы перегружен.

Задача, за которую взялись шестиклассники, отнюдь не простая. Как вы думаете, что должно стоять в ячейках таблицы на пересечении номеров автобусов и классов? Количество человек? Но если класс целиком садится в один автобус, оно и так нам известно.

Немного отвлечемся от автобусной экскурсии и сформулируем одно из самых главных правил информатики:

Любая задача, связанная с обработкой информации, требует в первую очередь осмысления связи между исходными данными и результатами.

Итак, что же общего имеют между собой первый автобус и, скажем, 7-й “А” класс? Ну конечно же 7-й “А” класс либо едет в этом автобусе, либо нет. Стало быть, и в соответствующей ячейке должны стоять либо единичка, либо нолик. Ну а для контроля наполняемости автобуса необходимо добавить еще одну строку (см. табл. 2):

Таблица 2

Класс	Едут на экскурсию	Первый автобус	Второй автобус	Третий автобус	Четвертый автобус
6-й “А”	23	1			
6-й “Б”	17		1		
6-й “В”	14				
6-й “Г”	7	1			
7-й “А”	18				1
7-й “Б”	5				
7-й “В”	19				
7-й “Г”	22				
8-й “А”	21				1
8-й “Б”	11				
8-й “В”	15				
<b>Итого в автобусе:</b>		30	17	0	39

Можете считать, что перед вами электронная таблица, трудиться над которой уже начали смысленные шестиклассники.

Естественно, что пересчет количества людей в автобусе производится автоматически, то есть у этих ячеек заполнен подвальный этаж. А как он заполнен, вы, наверное, уже догадались. Выполняя лабораторную работу, вы сумеете рассадить школьников по автобусам.

## Лабораторная работа № 4. Перевозка грузов

Перед диспетчером компании “Пануралтрансисбсервис” встала непростая задача. Три грузовика компании должны забрать с разных предприятий Екатеринбурга различные партии груза и доставить его в Тюмень.

Грузоподъемность каждой машины — 12 тонн, и хотелось бы распределить весь груз примерно поровну. Ну, быть может, допустив 100–150 кг перегрузки какого-нибудь грузовика. И в этом случае диспетчеру поможет электронная таблица.

Постараемся понять связь между строками и столбцами. И здесь есть отличие от предыдущего задания. А именно: вовсе необязательно, например, грузить весь кабель на одну машину. Поэтому в ячейках количества груза могут стоять не одни только нолики и единички.

В данном случае связь заключается в том, сколько единиц данного груза (упаковок, штук, ящиков) берет конкретная автомашинка.

Есть и еще одно отличие. Решите-ка такую задачку:

На складе было 11 станков. Четыре увез первый грузовик. Три — второй. А остальное увез третий. Спрашивается: сколько станков увез третий грузовик?

Что, задачка для первого класса? Ну так пусть компьютер сам ее и решает. А для этого необходимо заполнить подвальный этаж столбца, соответствующего третьему грузовику. Эти ячейки выделены светло-коричневым.

Для того чтобы нижняя формула не была такой громоздкой, как в случае рассадки школьников по автобусам, посчитаем вначале вес каждого груза в машине, заполнив верхнюю ячейку в розовых столбцах и продлив формулу вниз.

Ну и, разумеется, **требуется нижняя строка, которая контролирует загрузку каждой из машин. Ее подвальный этаж** (желтые ячейки) заполняется совсем уж просто: это автосумма розовых столбцов. Под конец красиво оформите таблицу так, чтобы с ней было удобно и приятно работать. Для этого расставьте границы и раскрасьте разными цветами столбцы и строки.

	A	B	C	D	E	F	G	H	I	J
1				Первый грузовик		Второй грузовик		Третий грузовик		
2	Наименование оборудования	Количество	Вес единицы груза (кг)	Кол-во	Вес	Кол-во	Вес	Кол-во	Вес	
3	Станки (штуки)	11	850							
4	Трубы (упаковка)	4	1930							
5	Буровое оборудование (ящики)	2	1700							
6	Отделочный камень (ящики)	4	1250							
7	Промышленны е электромоторы (штуки)	7	730							
8	Кабель (бухты)	5	1100							
9	Всего груза в машине (кг):									
10										
11										

Таблица распределения грузов

Желаем успешно поработать новоявленным диспетчером!

## Лабораторная работа № 5. Ведомость успеваемости, или Работа со стандартными формулами

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		Алгебра	Геометри	Русский	Литерату	Информа	Немецки	Физика	Физкультура	Двоек	Троек	Четверок	Пятерок
2	Абакулова Али	3	3	4	4	3	4	4	5				
3	Баранов Арсени	4	4	3	3	3	4	5	3				
4	Власова Екатери	3	5	4	5	4	4	5	3				
5	Волочугина Али	4	3	4	5	4	4	4	4				
6	Гилеб Александ	5	5	5	3	4	3	3	4				
7	Гречихо Кристи	5	5	4	5	5	5	5	5				
8	Едигарьев Нико	4	4	3	4	5	4	4	4				
9	Загороднов Иль	2	2	3	3	3	4	3	4				
10	Киселева Елиза	5	5	5	5	4	5	5	5				
11	Козлов Алексей	4	4	5	4	3	4	4	5				
12	Мартынова Екат	4	4	4	5	3	4	5	4				
13	Челуштанова Ан	5	5	5	5	5	5	5	5				
14	Шелегина Алеку	4	5	5	5	5	5	5	5				
15	Яценко Иван	2	3	4	3	3	3	4	5				
16													
17	Двоек									Отличников:			
18	Троек									С 1-й четверкой			
19	Четверок									С 1-й тройкой			
20	Пятерок									Не успевают:			
21										На 4 и 5			
22	СОУ												
23	Успевают												
24													
25													

Ведомость успеваемости учеников

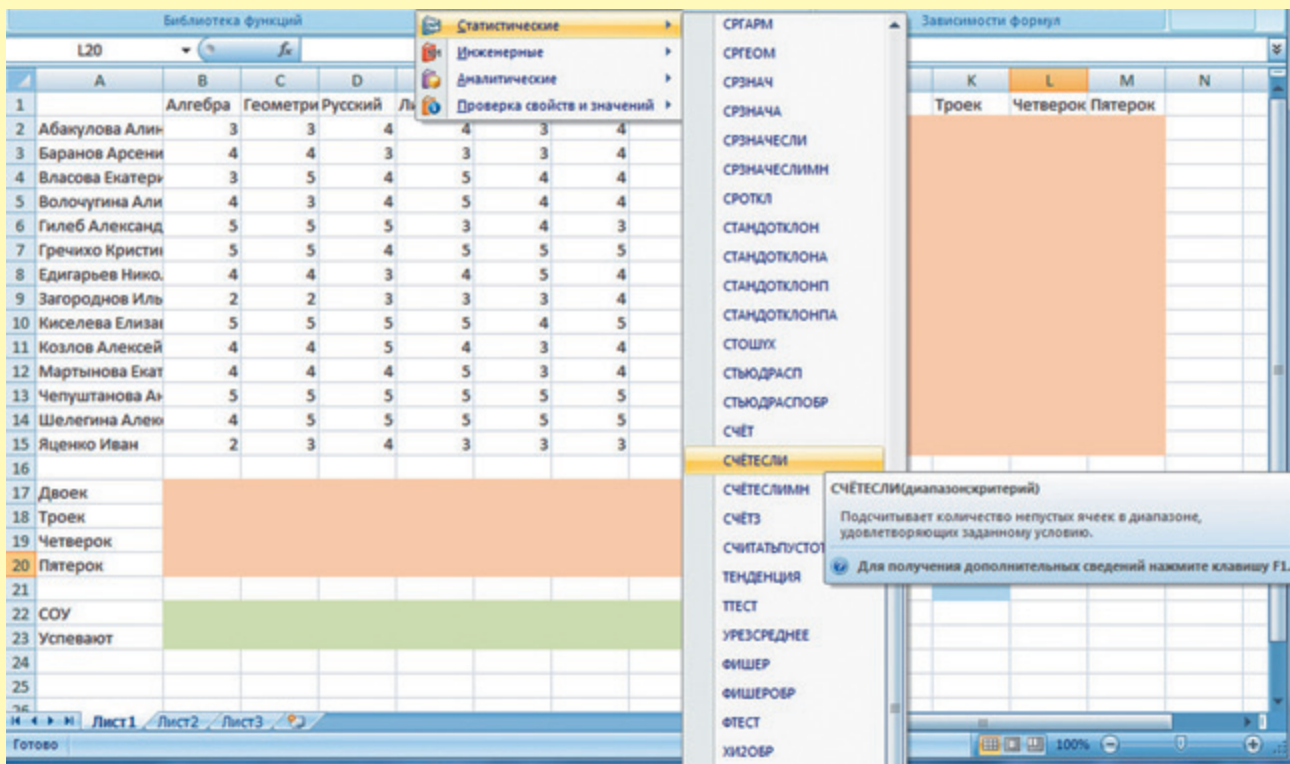
Перед вами — ведомость с оценками учеников 11-го класса.

Требуется выполнить следующие задания:

1. Поскольку такая ведомость составляется каждую четверть (каждые полгода в старших классах), хотелось бы, чтобы в нее вносились только оценки, а количество двоек, троек, четверок и пятерок вычислялось автоматически.

Для этого воспользуйтесь стандартной функцией СЧЁТЕСЛИ.





Выбор функции СЧЁТСЛИ

В качестве диапазона надо мышкой выделить строку или столбец, в которых считаем соответствующие оценки, в качестве критерия — просто указать значение.

Продлите функции вниз или вправо (при расчете итогов по предметам) и заполните рассчитываемыми данными розовые области.

2. Точно так же, с помощью этой же функции посчитайте количество отличников, тех, кто учится на 4 и 5 (будем считать, что у человека без троек двоек тоже нет), у кого одна четверка (считаем, что остальные — пятёрки) и одна тройка (то, что остальные двойки, тоже исключим). Иными словами, заполните формулами синюю область.

3. Для заполнения первой строки зеленой области воспользуемся формулой расчета степени обученности учащихся (СОУ) по каждому предмету:

$$\text{СОУ} = (\text{кол-во "5"} \times 1 + \text{кол. "4"} \times 0,64 + \text{кол. "3"} \times 0,36 + \text{кол. "2"} \times 0,14) / \text{кол-во учащихся.}$$

Это число нужно выразить в процентах. Для этого выделите все ячейки зеленой области, щелкните правой клавишей мыши и в пункте меню “Формат ячеек” выберите “Процентный”.

4. Точно так же посчитайте процент успеваемости по каждому предмету.

5. И под конец вставьте в ведомость гистограмму, показывающую, по какому предмету выше степень обученности.

## Лабораторная работа № 6.

### Оптовая покупка. Использование функции ЕСЛИ

Предположим, что наш магазин решил закупить у производителя утепленную одежду для спортивного зимнего сезона.

По условиям производства, производитель использует оптовую цену только в том случае, когда количество заказанного товара — не менее пяти штук.

Запишите в первую ячейку розового столбца формулу, позволяющую вычислить стоимость соответствующего товара в зависимости от количества в заказе, и продлите формулу вниз. В желтую ячейку добавьте общую стоимость всего заказа.

Иными словами, мы заполняем серую область, а в розовой и желтой сразу же происходит перерасчет.

Для выполнения этой работы вам потребуется функция ЕСЛИ, находящаяся в закладке “Логические функции”. Критерий очень простой: количество товара должно быть больше 4. Он так и записывается:  $> 4$ .

Надеемся, для вас не составит труда записать формулу расчета стоимости, если товара заказано от 1 до 4 штук, и больше 4 штук. Обратите внимание, что в рамках функции при записи формул знак “=” писать не надо.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	ОДЕЖДА ИЗ ФЛИСА						Розница	Опт			Кол-во	Цена		
2	Брюки ПОЛАР	Флис 200					1520	1170						
3	Брюки ТЕРМО	термопалотно					780	600						
4	Брюки СТРЕЙЧ	эластичное термобелье					950	730						
5	Брюки СТРЕЙЧ детские	эластичное термобелье					650	500						
6														
7	Куртка ХАСКИ	Виндблок					2415	1860						
8	Куртка ГОРИЗОНТ	Экстремальный и зимний спорт, материал - флис 200					2210	1700						
9	Куртка ГРАВИТИ						2300	1770						
10	Куртка ПАЛОМА	Активный отдых и зимний спорт, материал - флис 200, женская					1860	1430						
11	Куртка ФОКУС						2760	2120						
12	Куртка УЭЛЕН	Экстремальный и зимний спорт, материал-флис 200					1970	1375						
13														
14	Толстовка КЕНГА						1530	1180						
15	Толстовка ЭЛИТ	Экстремальный и зимний спорт, материал - флис 200					1490	1150						
16	Толстовка ЮКОН-К (200)	Экстремальный и зимний спорт, материал - флис 200					1450	1115						
17	Толстовка ЮКОН-К	Экстремальный и зимний спорт, материал-флис 100					1230	950						
18														
19	Футболка ТЕРМО	термопалотно короткий рукав					750	575						
20	Футболка ТЕРМО	термопалотно, длинный рукав					820	630						
21	Футболка СТРЕЙЧ	эластичное термобелье					1250	960						
22	Футболка СТРЕЙЧ детская	эластичное термобелье					780	600						
23														
24									Итого:					
25														

Таблица стоимости продукции

## Лабораторная работа № 7. Оформление прайс-листа

Посмотрите два прайс-листа (файлы Мегалекс.xls и Веломир.xls имеются на диске к этому номеру). Один, составленный компьютерной фирмой “Мегалекс”, группирует все товары по разделам, что делает удобным работу с ним. Другой, составленный явно менее компьютерно-грамотными менеджерами веломагазина, довольно тяжело воспринимать, поскольку никакой группировки в нем не использовано.

Помогите веломагазину привести свой прайс-лист в нормальный вид. Для этого товары нужно сгруппировать следующим образом:

При открытии прайс-листа мы должны видеть только следующие строки:

- Велоаксессуары
- Велозапчасти
- Велозащита, велошлемы
- Велоинструмент, велосмазки
- Велосипеды
- Запчасти для БМХ
- Очки велосипедные
- Питание

В свою очередь, при открытии велоаксессуаров должны появляться строки:

- Автобагажники
- Багажники, сумки
- Замки
- Компьютеры
- Крылья
- Фары, зеркала
- Фляги, флягодержатели

Велозапчасти содержат:

- Вилки, амортизаторы
- .....
- Цепи

То есть велоаксессуары и велозапчасти — это списки с двойной вложенностью.

Самое сложное — велосипеды.

Вначале должны открыться года:

- 2005
- 2006
- 2007
- 2008



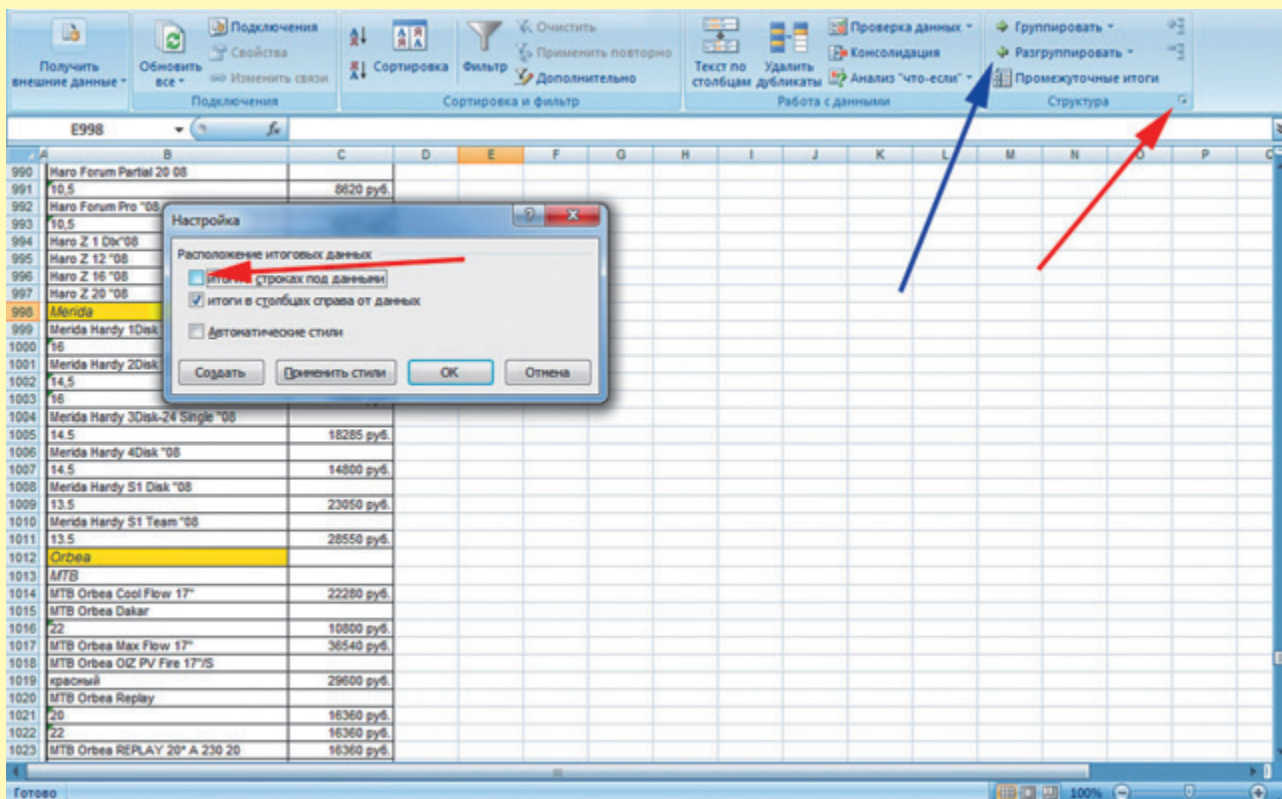
Если мы открываем год, должен открыться список марок. Например, для 2008-го:

- Atom
- Haro
- Merida
- Orbea
- Детские
- Прогулочные
- Шоссейные

И только потом уже конкретные товары.

Все остальные списки (велозащита, велоинструмент и др.) имеют одинарную вложенность. То есть с ходу открывается список соответствующих товаров.

Для выполнения этой работы в закладке “Данные” выберите выпадающее меню у слова “Структура” (см. красные стрелочки на рисунке) и уберите галочку перед словами “Итого в строках под данными”. В дальнейшем группируйте данные (или разгруппируйте их в случае ошибки) с помощью пунктов меню, указанных синими стрелочками. Чтобы не запутаться, группировку надо начинать с самого нижнего уровня.



Необходимые элементы управления для группировки данных

## Послесловие

В заключение — несколько методических замечаний по предложенному мини-курсу.

Давным-давно, в 1999 году, когда информатика, можно сказать, только входила, как предмет, в жизнь школы, я позволил себе пофантазировать о будущем:

“Представьте на долю секунды, что в вашем распоряжении все то изобилие техники, которое имеет сейчас американская школа. Как бы вы им распорядились?..

Нет, само собой, в кабинете (двух, трех) информатики появилась бы суперсовременная техника, которая дала бы наконец возможность перейти к непосредственному изучению передовых офисных технологий”.

Написано это было с изрядной долей сарказма. Кто же мог подумать, что информатика и информационные технологии и впрямь вырождаются в офисное “кнопконажимательство”. Поэтому хотелось бы отметить несколько принципиальных, на мой взгляд, вещей.

1. Современные офисные пакеты — весьма сложный программный продукт, изучая который можно с успехом освоить весь образовательный минимум по информатике и

информационным технологиям. Но это вовсе не означает, что только ими и надо заниматься.

2. Но даже занимаясь офисными пакетами, необходимо руководствоваться концепцией А.Г. Кушниренко, предполагающей, что информатика и информационные технологии должны развивать мышление. Поэтому лабораторные работы, подобные процитированной во введении, до определенной степени антипедагогичны.

3. Хотелось бы отметить пропедевтический характер предложенных лабораторных работ, когда школьники 6-го класса фактически начинают знакомиться с условными операторами, иерархической системой (в оформлении прайс-листа — Лабораторная работа № 7), а также (подспудно) с массивами и циклами. В этой связи особенно хочется отметить работу № 6 “Оптовая покупка”. Следует обратить особое внимание, чтобы она, по возможности, выполнялась самостоятельно и без подсказок, поскольку при пошаговом выполнении (с описанной подробно каждой операцией) правильно ввести одну функцию и продлить ее вниз займет не более пяти минут. Но именно личный опыт в данном случае бесценен.

Надеюсь, предложенные лабораторные работы тоже покажутся вам интересными и полезными.



# Дело в шляпе

А.В. Павлоцкий,  
Москва

Мир, который нас окружает, красив и многогранен. Но главное в том, что он — объемный. Как же отобразить этот объемный и красивый мир на экране компьютера? Как построить трехмерный объект? В этой статье мы рассмотрим два достаточно простых, но вполне пригодных для практических применений алгоритма компьютерной графики.

Стоп. Почему же, спрашивается, мы назвали статью “Дело в шляпе”? При чем тут шляпы и прочие элементы гардероба? Нам кажется, что тот, кто дочитает статью до конца и посмотрит все иллюстрации, сможет сам ответить на этот вопрос. Так что, впереди интрига.

## Что такое трехмерное пространство

Трехмерное пространство... Звучит как строка из фантастического рассказа... Ну очень похоже. Однако трехмерное пространство окружает нас со всех сторон. И фактически двумерное пространство или плоскость — это упрощение окружающего нас мира. Геометрия на плоскости известна нам с 7-го класса школы, поэтому наша задача будет заключаться в том, чтобы понять, как же работать с новым (а может быть, старым?) для нас третьим измерением и

как отображать на экране компьютера (на плоскости) объекты реального мира.

Итак, говоря языком математики, любая точка в трехмерном пространстве описывается с помощью уникального набора трех координат:  $x$ ,  $y$  и  $z$ . Точка с координатами  $(0; 0; 0)$  называется *началом координат* и обычно обозначается буквой “ $O$ ”. Также необходимо задать три взаимно перпендикулярных оси, которые пересекаются в начале координат. Назовем эти оси  $Ox$ ,  $Oy$  и  $Oz$ . Направление осей в принципе может быть произвольным, и его выбирают исходя из решаемой задачи. Таким образом, описанные оси образуют в пространстве три плоскости:  $XOZ$ ,  $YOZ$  и  $XOY$ .

Для определенности условимся, что экран компьютера представляется плоскостью  $XOZ$ , а ось  $Oy$  перпендикулярна экрану и смотрит на нас, зрителей.

Как же получить картинку объекта на экране компьютера? Опять возвращаясь к языку математики, можно ответить — **спроецировать** (то есть найти проекции) все точки объекта на экран (а как мы помним, экран для нас — это плоскость  $XOZ$ ).

Понять, как получается такая проекция, можно, мысленно проведя следующий опыт. Представьте, что вы смотрите через стекло на некоторый объект, а затем берете лазерную указку и через это



же стекло обводите контуры объекта. В результате на стекле будет выжжен рисунок — проекция объекта.

## Проекция точек и с чем их едят

Строго говоря, способов найти проекции, точки в пространстве на какую-либо плоскость — большое множество. Все зависит от того, насколько реалистичную картинку вы хотите получить. Это и ортогональные проекции, и перспективные, центральные и параллельные. Поскольку цель нашей статьи не исследование тонкостей математического аппарата для получения проекций, а рассмотрение конкретных алгоритмов, то мы остановимся на последнем методе более подробно, а любознательным читателям позволим самостоятельно изучить интересующий их вопрос.

Суть параллельного проецирования в следующем:

1) Мы смотрим на объект (а в простейшем случае — на точку) в пространстве как бы через плоское стекло, которое одновременно является нашим экраном;

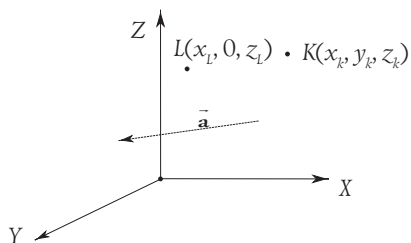
2) Причем на объект мы смотрим вдоль заданного вектора  $\vec{a}$ ;

3) Пересечение прямой, проходящей через точку объекта и параллельной вектору с плоскостью стекла (или экрана, или плоскости  $XOZ$ ), и есть проекция искомым точки, которую мы ищем.

Теперь приведем математический вывод и получим формулы параллельного проецирования, которые и будем использовать в дальнейшем.

Нам даны точка  $K(x_k, y_k, z_k)$  и вектор  $\vec{a} = \{x_a; y_a; z_a\}$ .

Необходимо найти координаты точки  $L(x_l, y_l, z_l)$ , которая является образом точки  $K$  при проецировании ее на плоскость  $ZOX$  (то есть  $y_l = 0$ ).



Найдем уравнение прямой, проходящей через точку  $K$  и параллельной вектору  $\vec{a}$ . Для этого воспользуемся каноническим уравнением прямой в пространстве:

$$\frac{x - x_k}{x_a} = \frac{y - y_k}{y_a} = \frac{z - z_k}{z_a}$$

Подставив в это уравнение координаты точки  $L$  и немного его упростив, получим, что:

$$\begin{cases} x_l = x_k - x_a * \frac{y_k}{y_a} \\ z_l = z_k - z_a * \frac{y_k}{y_a} \end{cases}$$

Давайте проанализируем наш результат. По заданному вектору  $\vec{a}$  для каждой точки трехмерного объекта мы можем найти координаты его проекции на плоскость  $ZOX$ , которая, в свою очередь, является нашим экраном, поэтому координата  $z$  плоскости соответствует его ординате.

Вот теперь, когда мы умеем проецировать трехмерный объект на экран, давайте пристально посмотрим на то, как можно описывать трехмерные объекты.

## Математическое описание трехмерных объектов

Вообще говоря, трехмерные объекты с помощью языка математики описываются точно так же, как и их двумерные собратья. Таких способов множество: словесный, параметрический, функциональный, алгоритмический и т.д. Мы остановимся на функциональном способе, оставив исследование остальных на откуп любознательному читателю.

Итак, для описания объекта мы должны задать функцию его определяющую. Вспомним, каким образом мы задаем функцию для двумерного объекта. Например, мы хотим описать параболу. Для этого мы запишем, что  $y = f(x) = x^2$ . Такая запись означает, что значение измерения  $y$  зависит от измерения  $x$  по определенному закону. Мы описали *функцию одной переменной*.

Другие примеры функций одной переменной — это:

$$\begin{aligned} 1) y &= f(x) = \sin(x); \\ 2) y &= f(x) = \ln(x); \end{aligned}$$

$$3) y = f(x) = e^{-\frac{x^2}{2}}.$$

Графиками представленных функций будут являться соответствующие им кривые на плоскости.

Что же пространство? В этом случае нам необходимо задать *функцию двух переменных*. Тогда измерение  $z$  будет зависеть от измерений  $x$  и  $y$ .

Приведем примеры:

$$1) z = f(x, y) = \sqrt{x^2 + y^2};$$

$$2) z = f(x, y) = \cos(\sqrt{x^2 + y^2});$$

$$3) z = f(x, y) = xy(x - y) + \frac{x + y}{\sqrt{x^2 + y^2}}.$$

Графиками же этих функций будут являться **поверхности** в пространстве, которые будут выглядеть примерно так:

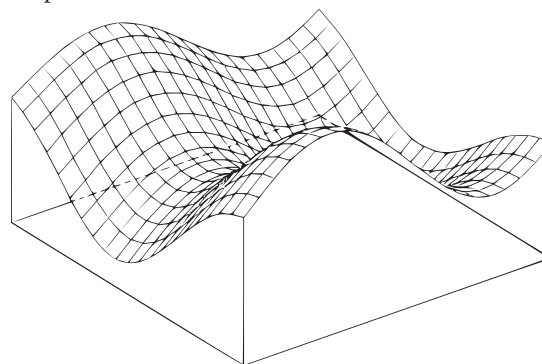


Рис. 1. Как выглядит поверхность?

Согласитесь, что первая возникающая ассоциация с приведенным рисунком — это кусок ткани, наброшенный на некоторую опору. Изначно, правда? Так вот, эта

опора — не что иное, как часть плоскости (в нашем случае  $XOY$ ), и каждой точке этой плоскости соответствует точка на поверхности.

Итак, для того чтобы построить на экране компьютера реальный трехмерный объект, мы должны:

1) описать его на языке математики с помощью функции двух переменных (ну, или хотя бы максимально приблизиться к этому ☺);

2) придумать алгоритм, который по заданной формуле построит наш объект.

Вот этим мы сейчас и займемся.

## Алгоритм “художника” для построения трехмерных объектов

Первый алгоритм, который мы рассмотрим, называется *алгоритм художника*. Нам неизвестна причина такого названия, его история, а также назван ли он в честь какого-то определенного художника или посвящен всем художникам мира. Да и к изобразительному искусству он в принципе имеет слабое отношение. Так что оставим эти изыскания и обратимся к сухим фактам ☺.

Вспомним, что мы строим функцию двух переменных  $f(x, y)$ , причем набор точек с координатами  $(x, y)$  описывает часть плоскости  $XOY$ . Разобьем эту плоскость на квадраты стороны  $step$ , проведя соответствующие прямые, параллельные осям  $OX$  и  $OY$ . Получим своего рода сетку. Теперь для каждой точки квадрата мы будем искать значение функции  $f(x, y)$  — координату  $z$  точки на поверхности. Получается, что квадрат с плоскости  $XOY$  превращается в четырехугольную фигуру на поверхности. А теперь мы спроецируем эту фигуру на экран (то есть на плоскость  $XOZ$ ) вдоль вектора  $\vec{a}$  (ведь мы же как-то смотрим на наш объект ☺).

В итоге квадрат с плоскости  $XOY$  трансформируется в четырехугольник на плоскости  $XOZ$ . Вот в этом месте нашего рассказа необходимо сделать два замечания:

1) Почему мы стали разбивать плоскость именно на квадраты? Внимательный читатель скорее всего заметил, что на самом-то деле нам достаточно работать с треугольниками, ведь плоскость однозначно определяется тремя точками в пространстве. Да, именно так. Но не надо забывать тот факт, что мы изучаем данный алгоритм с точки зрения прикладного программирования. А в этом случае, нам кажется, работать с квадратами проще и удобнее. Хотя можно работать и с треугольниками, и даже с пятиугольниками.

2) Говоря об итоговой проекции квадрата на ось  $XOZ$ , мы определили ее как **четыреугольник**. Однако может случиться такая ситуация, что проекция трансформируется в треугольник, отрезок и даже точку. Однако с точки зрения программирования это не важно, поскольку мы подходим к построению формально: строим проекции четырех точек и соединяем их в том порядке, в котором просматриваем. Что в этом случае получится — не важно. Главное тут, что пятиугольник не получится никогда ☺.

А вот теперь начинаются “художества”.

Начнем рисовать наш объект с самого дальнего от нас наблюдателя, участка. Под участком мы будем понимать

один из квадратиков, на которые поделена плоскость  $XOY$ . Причем определимся, что рисовать мы будем слева направо. В принципе такое поведение не обязательно, но *отправиться все-таки стоит*. Кроме того, каждый четырехугольник мы будем закрашивать белым цветом.

Напомним также, что в итоге мы строим проекции каждого квадратика на плоскость  $XOZ$ .

Итак, мы нарисовали проекцию самого дальнего слоя (см. рис. 2).

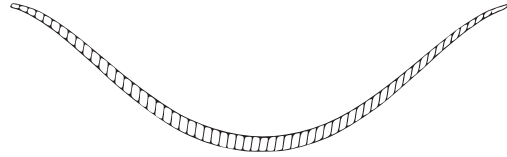


Рис. 2. Мы построили “дальний” слой

Теперь начнем строить слой, который находится “ближе” к наблюдателю, который, напомним, смотрит на объект вдоль вектора  $\vec{a}$ . Причем, рисуя новые четырехугольники (и закрашивая их белым цветом), мы совершенно осознанно будем “затирать” старые. В этой, на первый взгляд кажущейся, “жестокости” и скрывается изюминка алгоритма. Как бы мы не смотрели на поверхность, ближние участки **будут обязательно закрывать** более дальние от нас. Именно поэтому мы можем не задумываясь “закрывать” ранее нарисованные участки, если такое понадобится. На следующих рисунках приведены некоторые стадии построения поверхности:  $z = f(x, y) = 100 * \cos\left(\frac{\pi}{180} \sqrt{x^2 + y^2}\right)$ .

Вот она — наша шляпа! (Рис. 3 на с. 21.)

Приведем фрагмент программы, которая строит поверхность. Программа написана на языке Delphi, а ее полная версия приводится на диске, сопровождающем данный номер газеты.

Функция **Surface** — по номеру  $n$  и координатам  $x, y$  возвращает  $z$ -координату точки поверхности с номером  $n$ .

Функции **Xs** и **Ys** служат для перевода координат из декартовой системы в “экранную”.

**Dy** и **Dx** — определяют размеры плоскости.

**Stepx** и **stepy** — шаги сетки по осям  $OX$  и  $OY$

**procedure** TSurfaceForm.BuildSurface( $n$ : Integer);

```
var
  x, y: integer;
  z1, z2, z3, z4: real;
  x1s, x2s, x3s, x4s, z1s, z2s, z3s, z4s: real;
begin
  // Начинаем идти вдоль оси OY от -dy до dy
  // с шагом stepy
  y := -dy;
  while y <= dy - stepy do
  begin
    // Начинаем идти вдоль оси OX от -dx до dx
    // с шагом stepx
    x := -dx;
    while x <= dx - stepx do
    begin
      // Вычисляем координаты z четырех точек
      // квадрата
      z1 := surface(n, x, y);
      z2 := surface(n, x, y + stepy);
      z3 := surface(n, x + stepx, y + stepy);
      z4 := surface(n, x + stepx, y);
      // Находим их проекции на XOZ по формулам
      // параллельного проецирования
```



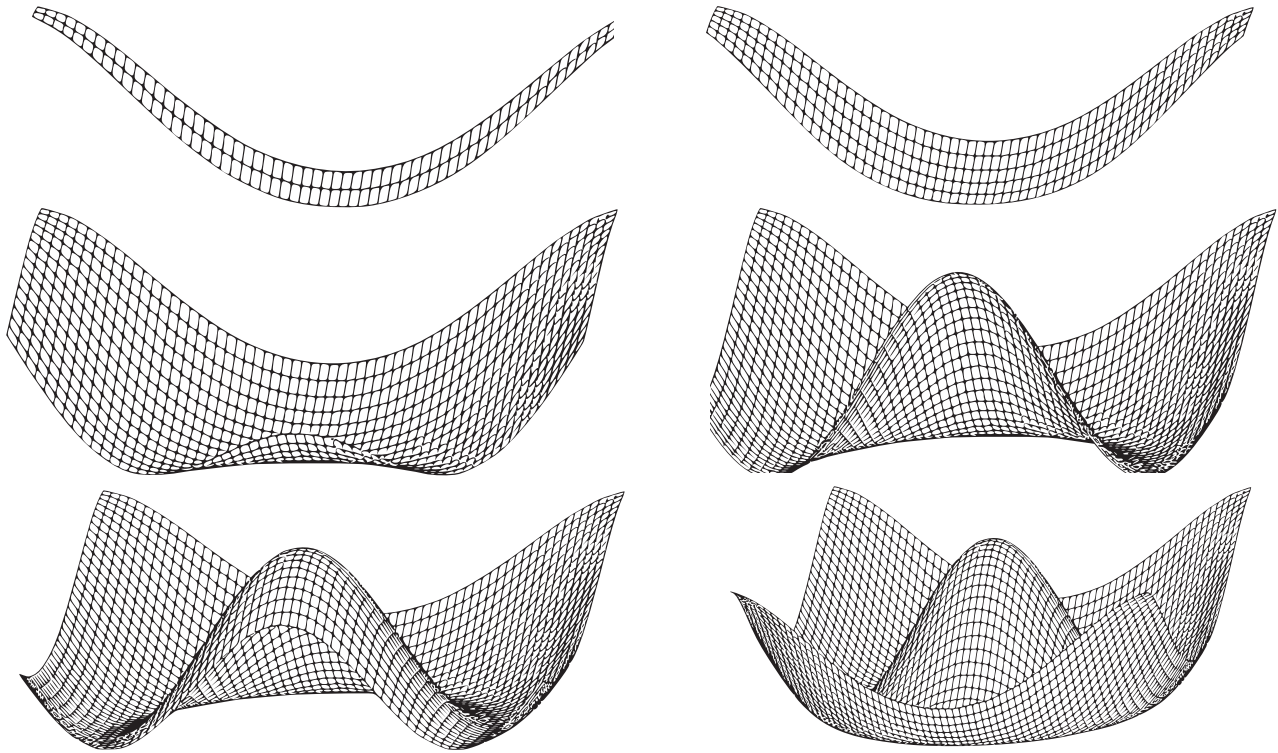


Рис. 3. Стадии построения поверхности методом художника (2, 5, 15, 30, 40, полностью)

```

x1s := x - xa*y/ya;
z1s := z1 - za*y/ya;
x2s := x - xa*(y+stepy)/ya;
z2s := z2 - za*(y+stepy)/ya;
x3s := x + stepx - xa*(y+stepy)/ya;
z3s := z3 - za*(y+stepy)/ya;
x4s := x + stepx - xa*y/ya;
z4s := z4 - za*y/ya;
// Строим четырехугольник и заливаем
// его белым цветом
PaintBox1.Canvas.Polygon([Point(Xs(X1s),
                                Ys(Z1s)),
                            Point(Xs(X2s),
                                Ys(Z2s)),
                            Point(Xs(X3s),
                                Ys(Z3s)),
                            Point(Xs(X4s),
                                Ys(Z4s))]);

// Идем в следующую точку x
x := x + stepx;
end;
// Идем в следующую точку y
y := y + stepy;
end;
end;

```

### Алгоритм “плавающего” горизонта

Другой алгоритм построения трехмерных объектов, который мы рассмотрим, носит название “метод плавающего горизонта”, или “минимаксный метод”. Суть метода заключается в следующем:

1) Разделим плоскость  $XOY$  на полосы, вдоль оси  $OX$  (напомним, что в предыдущем методе мы делили плоскость на квадраты) с некоторым шагом. Получим множество прямых, параллельных оси  $OX$ . *Кстати, можно также произвести раздел и вдоль оси  $OY$ . Суть метода от этого не поменяется.*

2) Будем пробегать по каждой прямой от начала до конца, находить значение функции  $z = f(x, y)$ , а затем

получать проекцию точки  $z$  на плоскость  $XOZ$ , применяя знакомое нам параллельное проецирование.

Если мы больше ничего не предпримем, то получим на экране набор пересекающихся кривых, очень отдаленно напоминающий шляпу из предыдущего раздела (см. рис. 4).

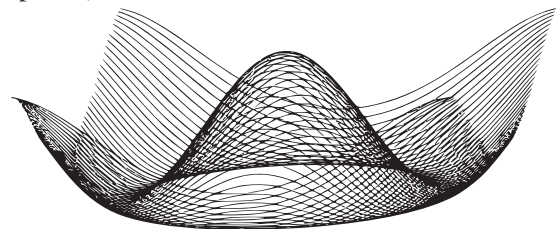


Рис. 4. Жуткая шляпа

Поэтому нам необходимо удалить те линии, которые являются невидимыми для нас, наблюдателей. Для этого заведем массив записей длины, равной количеству пикселей оси  $OX$ . (Мы, конечно, понимаем, что в математике на оси бесконечное количество точек, поэтому и употребили слово “пиксель”.) Описать этот массив на языке Паскаль можно примерно вот так:

```

Type
  TDoublePoint = Record
    Min, Max: double;
  End;
Var
  MinMaxArray: Array [0..1000] of TDoublePoint;

```

В этом массиве мы будем хранить для каждой точки  $x$  экрана минимальное и максимальные значения проекции координаты  $z$ , которые получались у нас в процессе вычислений. Этот момент стоит пояснить подробнее.

Проекция первой линии инициализирует массив своими значениями. Эта линия будет нарисована целиком. Для следующих линий будем придерживаться

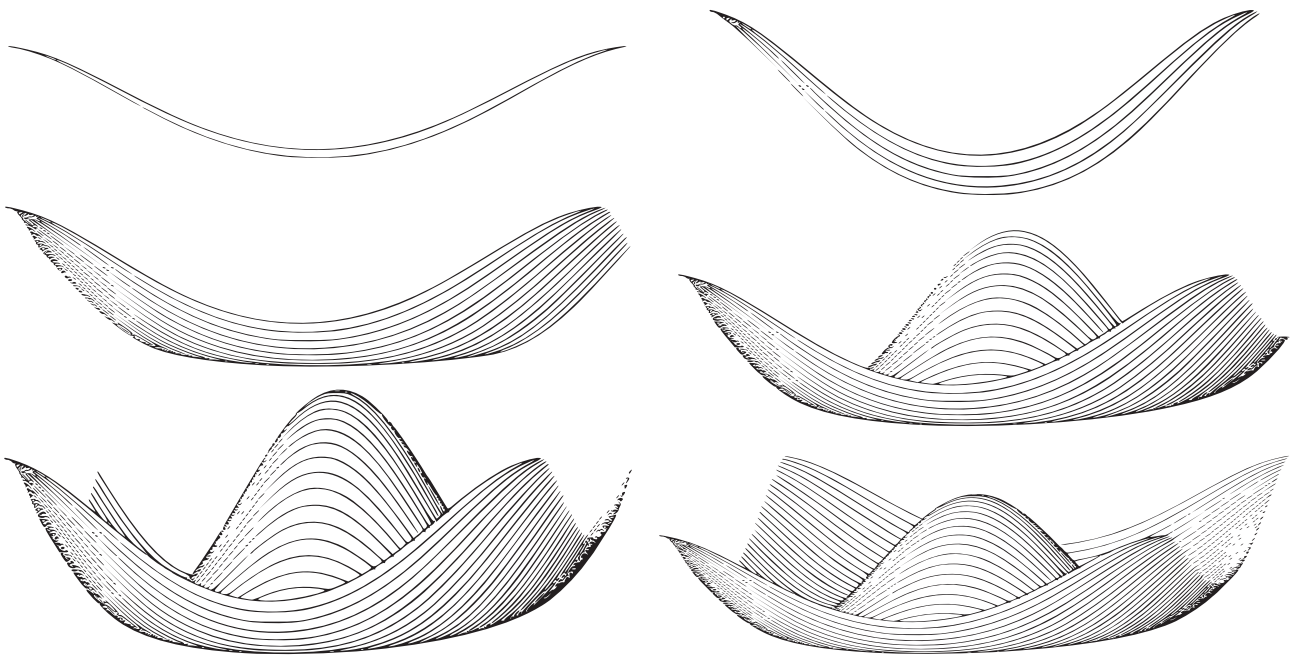


Рис. 5. Стадии построения поверхности методом плавающего горизонта (2, 5, 15, 30, 40, полностью)

такого правила: если координата  $z$  проекции попадает в интервал между минимумом и максимумом для  $x$ , то точка  $(x, z)$  не рисуется, в противном же случае рисуется. Если точка отрисовывается, то необходимо обновить значение или максимума, или минимума в массиве для координаты  $x$  этой точки.

Можно заметить, что вторая линия тоже будет нарисована целиком. На рис. 5 приведены стадии построения поверхности методом плавающего горизонта.

Поскольку в приведенном алгоритме используется массив, в котором хранятся минимумы и максимумы значений проекции координаты  $z$ , такой метод часто называют минимаксным. И еще на одну вещь мы хотим обратить внимание читателей. Во втором алгоритме мы строили поверхность **от себя и справа налево** в отличие от первого. Это сделано преднамеренно, для того чтобы продемонстрировать множество вариаций конечных реализаций того или иного алгоритма.

Приведем также фрагмент программы на языке Delphi, которая строит аналогичные поверхности методом плавающего горизонта.

```

procedure TForm1.BuildSurface(N: Integer);
var
  y: integer;
  z: double;
  x, xs_, zs: double;
begin
  // Заполним минимумы массива максимальными
  // возможными значениями,
  // а максимумы – минимально возможными
  for y := 0 to 1000 do
    begin
      MinMaxArray[y].Min := PaintBox1.Height;
      MinMaxArray[y].Max := 0;
    end;
  // Начинаем идти по полосам
  y := dy;
  while y >= -dy do
    begin
      x := dx;
      // вычисляем значение функции

```

```

      z := surface(n, x, y);
      // находим проекцию точки
      xs_ := x - xa*y/ya;
      zs := z - za*y/ya;
      // переходим в эту точку
      PaintBox1.Canvas.Moveto(Xs(xs_), Ys(zs));
      x := dx-0.3;
      // идем по всем точкам кривой
      while x >= -dx do
        begin
          // вычисляем значение функции
          z := surface(n, x, y);
          // находим проекцию точки
          xs_ := x - xa*y/ya;
          zs := z - za*y/ya;
          // заполняем минимальное и максимальное
          // значения
          MinMaxArray[xs(xs_)].Min :=
            min(MinMaxArray[xs(xs_)].Min, ys(zs));
          MinMaxArray[xs(xs_)].Max :=
            max(MinMaxArray[xs(xs_)].Max, ys(zs));
          // определяем, ставим ли мы точку или нет
          if (ys(zs) <= MinMaxArray[xs(xs_)].Min) or
            (ys(zs) >= MinMaxArray[xs(xs_)].Max)
          then
            PaintBox1.Canvas.Lineto(Xs(xs_), Ys(zs))
          else
            PaintBox1.Canvas.Moveto(Xs(xs_), Ys(zs));
          x := x - 0.3;
        end;
      y := y - stepy;
    end;
end;

```

## Закключение

В данной статье мы рассмотрели два интересных алгоритма, которые позволяют строить трехмерные поверхности. Мы надеемся, что эта публикация покажет вам, что, применяя несложные математические преобразования и базовые знания алгоритмизации, можно решать красивые задачи и получать эффектные результаты.



Газета для пытливых учеников  
и их талантливых учителей  
[vmi@1september.ru](mailto:vmi@1september.ru)



## ШКОЛА ПРОГРАММИРОВАНИЯ

### Обмены, обмены...

Продолжение. Начало см. "В мир информатики"  
№ 152, 155 ("Информатика" № 22/2010, 1/2011)

В этой части статьи мы рассмотрим особенности программ, связанных с обменом значениями элементов квадратного двумерного массива.

**9. Задача** "Элементы побочной диагонали квадратного двумерного массива размером  $n$  на  $n$  расположить в зеркальном порядке (побочная диагональ квадратного массива проходит от правого верхнего элемента к левому нижнему)"

Прежде всего вспомним задачу 3 из первой части статьи и установим, что и здесь в операторе цикла обмены должны проводиться для  $(n \operatorname{div} 2)$  пар элементов.

Каковы индексы этих элементов? Пусть  $n = 10$ . Составим такую таблицу:

Индексы элемента	Индексы элемента, с которым он меняется значениями
1, 10	10, 1
2, 9	9, 2
...	...
5, 6	6, 5

В общем виде эта таблица, из которой следует ответ на вопрос, имеет вид:

Индексы элемента	Индексы элемента, с которым он меняется значениями
$1, n$	$n, 1$
$2, n - 1$	$n - 1, 2$
...	...
$n \operatorname{div} 2, n \operatorname{div} 2 + 1$	$n \operatorname{div} 2 + 1, n \operatorname{div} 2$

Если номер строки элемента в левом столбце таблицы обозначить  $i$ , то второй индекс этого элемента будет равен  $n - i + 1$ . Тогда фрагмент программы, в котором решается задача, может быть оформлен так:

```
нц для i от 1 до div(n, 2)
  всп := a[i, n - i + 1]
  a[i, n - i + 1] := a[n - i + 1, i]
  a[n - i + 1, i] := всп
```

кц

— где всп — имя вспомогательной переменной для хранения одного из обмениваемых значений.



**10. Задача** "Поменять элементы главной и побочной диагоналей квадратного двумерного массива размером  $n$  на  $n$  (главная диагональ квадратного массива проходит от левого верхнего элемента к правому нижнему)"

Ясно, что здесь обмениваться значениями должны все  $n$  пар элементов. Индексы обмениваемых элементов:

Индексы элемента	Индексы элемента, с которым он меняется значениями
1, 1	1, $n$
2, 2	2, $n - 1$
...	...
$n - 1, n - 1$	$n - 1, 2$
$n, n$	$n, 1$

а соответствующий фрагмент программы:

```
нц для i от 1 до n
  всп := a[i, i]
  a[i, i] := a[i, n - i + 1]
  a[n - i + 1, i] := всп
```

кц

**11. Задача** "Дан квадратный двумерный массив размером  $n$  на  $n$ . Определить, как будет выглядеть этот массив при повороте его по часовой стрелке"

на 180°. “Ориентация” цифр по вертикали, естественно, меняться не должна”

Для наглядности сделаем рисунки (см. рис. 1 и 2), на основе которых составим таблицу:

Индексы элемента в исходном массиве	Новые значения индексов этого элемента
1, 1	10, 10
1, 2	10, 9
...	...
1, 9	10, 2
1, 10	10, 1
2, 1	9, 10
2, 2	9, 9
...	...
2, 9	9, 2
2, 10	9, 1
...	...

Из нее видно, что индексы нового положения элементов исходного массива определяются так:

1) новый номер строки зависит от номера строки исходного массива:  $i = n - i + 1$ ;

2) новый номер столбца зависит от номера столбца исходного массива:  $j = n - j + 1$ .

На основе полученных зависимостей новый массив можно получить с помощью следующего фрагмента программы:

```
нц для i от 1 до n
  нц для j от 1 до n
    ▪ b[i, j] := a[n - i + 1, n - j + 1]
  кц
```

кц

или такого:

```
нц для i от 1 до n
  нц для j от 1 до n
    ▪ b[n - i + 1, n - j + 1] := a[i, j]
  кц
```

кц

Видно, что в обоих случаях используется дополнительный массив  $b$ . А можно ли обойтись без дополнительного массива? Можно. Из рис. 1 и 2 можно заметить, что для решения задачи следует поменять местами элементы массива, расположенные над и под побочной диагональю, а затем и “половинки” этой диагонали. Вторую задачу мы уже решали. Обсудим первую.

На основе рис. 3, на котором элементы над побочной диагональю оттенены, можем составить таблицу:

Индекс элемента над побочной диагональю	С каким элементом он меняется значениями
1, 1	$n, n$
1, 2	$n, n - 1$
...	...
1, $n - 1$	$n, 2$
2, 1	$n - 1, n$
2, 2	$n - 1, n - 1$
...	...
2, $n - 2$	$n - 1, 3$
...	...
$n - 1, 1$	$2, n$

— с учетом которой обмен “половинок” массива проводится так:

```
нц для i от 1 до n - 1
  нц для j от 1 до n - i
    ▪ всп := a[i, j]
    ▪ a[i, j] := a[n - i + 1, n - j + 1]
    ▪ a[n - i + 1, n - j + 1] := всп
  кц
```

кц

1	2	3	...	8	9	10
11	12	13	...	18	19	20
21	22	23	...	28	29	30
...	...	...	...	...	...	...
71	72	73	...	78	79	80
81	82	83	...	88	89	90
91	92	93	...	98	99	100

Рис. 1. Исходный массив

100	99	98	...	93	92	91
90	89	88	...	83	82	81
80	79	78	...	73	72	71
...	...	...	...	...	...	...
30	29	28	...	23	22	21
20	19	18	...	13	12	11
10	9	8	...	3	2	1

Рис. 2. Полученный массив


Рис. 3. Элементы над побочной диагональю

**12. Задача** “Дан квадратный двумерный массив размером  $n$  на  $n$ . Определить, как будет выглядеть этот массив при повороте его по часовой стрелке на 90°. “Ориентация” цифр по вертикали, естественно, меняться не должна”

И здесь для наглядности сделаем рисунки (см. рис. 4 и 5). Из них видно, что индексы нового положения элементов исходного массива определяются так:

1) номер столбца равен номеру строки исходного массива;

2) номер строки  $i$  зависит от номера столбца  $j$  исходного массива:  $i = n - j + 1$ .

1	2	3	...	8	9	10
11	12	13	...	18	19	20
21	22	23	...	28	29	30
...	...	...	...	...	...	...
71	72	73	...	78	79	80
81	82	83	...	88	89	90
91	92	93	...	98	99	100

Рис. 4. Исходный массив

91	81	71	...	21	11	1
92	82	72	...	22	12	2
93	83	73	...	23	13	3
...	...	...	...	...	...	...
98	88	78	...	28	18	8
99	89	79	...	29	19	9
100	90	80	...	30	20	10

Рис. 5. Полученный массив

На основе полученных зависимостей новый массив можно получить с помощью следующего фрагмента программы:

```
нц для i от 1 до n
  нц для j от 1 до n
    ▪ b[i, j] := a[n - j + 1, j]
  кц
кц
```

или такого:

```
нц для i от 1 до n
  нц для j от 1 до n
    ▪ b[n - j + 1, j] := a[i, j]
  кц
кц
```

Видно, что в них используется дополнительный массив. А нельзя ли обойтись без него и здесь? Можно, но сделать это несколько сложнее, чем при решении предыдущей задачи. На основе рис. 4 и 5 можно сделать вывод о том, что из области 1 все элементы переходят в область 2 (см. рис. 6), из области 2 — в область 3, из области 3 — в область 4, из области 4 — в область 1.

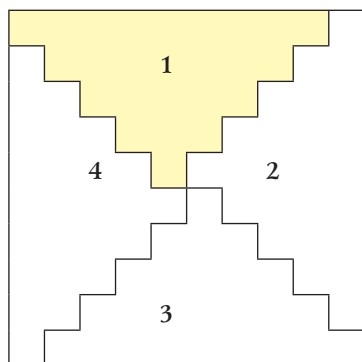


Рис. 6

Такие обмены можно реализовать, например, с помощью следующей схемы действий<sup>1</sup>:

- цикл** для каждого элемента области 1
- 1. Запоминаем его во вспомогательной переменной
  - 2. Записываем на его место соответствующий элемент области 4
  - 3. На место этого элемента области 4 записываем соответствующий элемент области 3
  - 4. На место этого элемента области 3 записываем соответствующий элемент области 2
  - 5. На место этого элемента области 2 записываем "запомненное" ранее значение
- конец цикла**

Соответствующий фрагмент программы:

```
| Рассматриваем все элементы в области 1
нц для i от 1 до div(n, 2)
  нц для j от i до n - i
    | Запоминаем каждый элемент
    всп := a[i, j]
    | Записываем элемент из области 4
    a[i, j] := a[n - j + 1, i]
    | Записываем элемент из области 3
    a[n - j + 1, i] := a[n - i + 1, n - j + 1]
    | Записываем элемент из области 2
    a[n - i + 1, n - j + 1] := a[j, n - i + 1]
    | Записываем "запомненное" значение
    a[j, n - i + 1] := всп
  кц
кц
```

**Примечание.** Предлагаем читателям самостоятельно убедиться в правильности приведенных зависимостей между индексами элементов различных областей.

## Задания для самостоятельной работы

1. Составить программу, в которой элементы главной диагонали квадратного двумерного массива размером  $n$  на  $n$  расположить в зеркальном порядке относительно исходного состояния.
2. Определить, как будет выглядеть заданный квадратный двумерный массив размером  $n$  на  $n$  при повороте его против часовой стрелки на  $90^\circ$ . "Ориентация" цифр по вертикали, естественно, меняться не должна. Дополнительный массив в программе не использовать.

## "ЛОМАЕМ" ГОЛОВУ

### Два sudoku

Решите, пожалуйста, две японские головоломки "судоку":

1) простую:

		6		4	3	1		9
7		8			1	4		
4	9			5				2
	8							7
	1		7		5		2	
2							4	
9							3	6
		2	4			7		8
8		3	9	7		5		

2) более сложную:

				6		1		9
		8		9			5	
5		9	2			7	6	
				4		3		8
	6	3		2	9			
4							9	
	3					4		
					5	9	8	
			9	7				

Ответы присылайте в редакцию (можно решать не все судоку).

<sup>1</sup> Похожая задача с обменом значениями трех переменных величин была предложена для самостоятельной работы в первой части статьи.



**1. 10 ребусов на одну тему**

Ответы. Ребус № 1 — ФОРМАТ. Ребус № 2 — ДОКУМЕНТ. Ребус № 3 — ТАБЛИЦА. Ребус № 4 — СЕРВИС. Ребус № 5 — ВСТАВКА. Ребус № 6 — ЛИНЕЙКА. Ребус № 7 — СПРАВКА. Ребус № 8 — КУРСОР. Ребус № 9 — СПИСОК. Ребус № 10 — РИСУНОК.

Правильные ответы прислали:

— Адамюк Анастасия, Грибанов Владлен, Дюбарова Анастасия, Клименко Надежда, Таскаева Вероника и Шошина Екатерина, г. Лесосибирск Красноярского края, поселок Стрелка, школа № 8 им. Константина Филиппова, учитель **Лопатин М.А.**;

— Акбарова Снежанна, Габдуллин Руслан, Зайнакаев Вадим, Коробейникова Полина, Матиенко Анна, Султанов Алмаз и Швецова Анна, Республика Башкортостан, Краснокамский р-н, село Николо-Березовка, школа № 1, учитель **Ситдикова А.Г.**;

— Аксенов Василий, Демьянова Елена, Костюнин Александр и Хомякова Анна, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Барабаш Анастасия, Бурцова Татьяна, Гавшина Анастасия, Горбунова Анастасия, Зуева Дарья, Савинов Сергей и Титовец Олеся, г. Белово Кемеровской обл., поселок Краснобродский, школа № 31, учитель **Зайцева Л.А.**;

— Бекезина Кристина, Вадьковская средняя школа, Брянская обл., Погарский р-н, учитель **Цыганкова И.Ю.**;

— Биякова Екатерина, Дыльченко Роман и Сурдушкин Иван, средняя школа села Ложниково, Омская обл., Тарский р-н, учитель **Коровин Д.В.**;

— Вылка Татьяна, основная школа поселка Каратайка, Архангельская обл., учитель **Безумова В.А.**;

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

— Градова Анна, Кононова Алина, Кобелева Анастасия, Толмачева Екатерина и Челишева Ольга, Кемеровская обл., поселок Малиновка, школа № 30 им. Н.Н. Колокольцова, учитель **Толмачева Н.В.**;

— Гриценко Любовь, Толмачев Владимир и Эннс Александр, г. Рубцовск Алтайского края, школа № 1, учитель **Толмачева Н.П.**;

— Данилов Влад, Иванова Ирина, Кириченко Анастасия, Комарова Анастасия, Кондратьева Яна и Селезнева Дарья, Смоленская обл., г. Демидов, школа № 1, учитель **Кордина Н.Е.**;

— Евгасьев Вячеслав, Кардубанова Кристина и Карзунов Михаил, Санкт-Петербург, г. Зеленогорск, школа № 445, учитель **Зорина Е.М.**;

— Егоров Евгений, Терентьев Руслан и Тимофеева Ульяна, основная школа села Именеве, Республика Чувашия, Красноармейский р-н, учитель **Тимофеева И.А.**;

— Желтышева Елизавета, средняя школа поселка Суксун, Пермский край, учитель **Желтышева Л.А.**;

— Игошев Константин, Свердловская обл., г. Ревда, школа № 10, учитель **Погуляй О.Д.**;

— Камисов Максим и Пасетов Алексей, Свердловская обл., Красноуфимский р-н, Тавринская средняя школа, учитель **Ярцев В.А.**;

— Каюмов Адель и Каюмов Ансель, Республика Татарстан, г. Бавлы, гимназия № 4, учитель **Шафиков Н.Р.**;

— Латышев Александр и Нухова Милауша, средняя школа села Сейтяково Балтачевского р-на, Республика Башкортостан, учитель **Загафуранова А.Ф.**;

— Тымченко Александр, Приморский край, г. Владивосток, школа № 22, учитель **Гниляк Т.В.**;

— Ядзевичюс Стасис, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**

**2. Числовой ребус с КУБом**

Напомним, что необходимо было решить числовой ребус  $КУБ = Б^{У+К}$ .

Решение

Прежде всего можно установить, что показатель степени  $(У + К)$  не может быть меньше трех, поскольку даже  $9^2$  — не трехзначное число, и не больше 9, так как  $2^{10} > 999$ .

После этого определим значения показателя степени, дающие трехзначное значение  $Б^{У+К}$ . Для этого может быть использована электронная таблица Microsoft Excel. В приведенном ниже фрагменте листа в диапазоне В3:G10 получены значения  $Б^{У+К}$ .

	А	В	С	Д	Е	F	G	Н
1	Показатель степени У + К							
2	Б	3	4	5	6	7	8	9
3	2	8	16	32	64	128	256	512
4	3	27	81	243	729	2187	6561	
5	4	64	256	1024	4096	16384	65536	
6	5	125	625	3125	15625	78125	390625	
7	6	216	1296	7776	46656	279936	1679616	
8	7	343	2401	16807	117649	823543	5764801	
9	8	512	4096	32768	262144	2097152	16777216	
10	9	729	6561	59049	531441	4782969	43046721	

Формула в ячейке В3 может быть оформлена так, что ее можно распространить (скопировать на остальные ячейки диапазона В3:G10). Вид такой формулы установите самостоятельно.

Допустимые варианты для анализа приведены в таблице:

Б	Показатель степени У + К						
	3	4	5	6	7	8	9
2					128	256	512
3			243	729			
4		256					
5	125	625					
6	216						
7	343						
8	512						
9	729						

Из нее видно, что значение цифры **Б** совпадает с последней цифрой результата в шести случаях:

- 1)  $512 = 2^9$ ;
- 2)  $243 = 3^5$ ;
- 3)  $125 = 5^3$ ;
- 4)  $625 = 5^4$ ;
- 5)  $216 = 6^3$ ;
- 6)  $729 = 9^3$ ;

— из которых с точки зрения значений **У** и **К** подходят только третий и пятый варианты.

Итак, ребус имеет два решения:

- 1) **Б** = 5, **У** = 2, **К** = 1 ( $125 = 5^{2+1}$ );
- 2) **Б** = 6, **У** = 1, **К** = 2 ( $216 = 6^{1+2}$ ).

Ответы прислали:

— Адамюк Анастасия, Грибанов Владлен, Дюбарова Анастасия, Клименко Надежда, Таскаева Вероника и Шошина Екатерина, г. Лесосибирск Красноярского края, поселок Стрелка, школа № 8 им. Константина Филиппова, учитель **Лопатин М.А.**;

— Акбарова Снежанна, Габдуллин Руслан, Зайнакаев Вадим, Коробейникова Полина, Матиенко Анна, Султанов Алмаз и Швецова Анна, Республика Башкортостан, Краснокамский р-н, село Николо-Березовка, школа № 1, учитель **Ситдикова А.Г.**;

— Аксенов Василий, Демьянова Елена, Костюнин Александр и Хомякова Анна, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**;

— Биякова Ирина и Курманова Алина, средняя школа села Ложниково, Омская обл., Тарский р-н, учитель **Коровин Д.В.**;

— Вылка Татьяна, основная школа поселка Каратайка, Архангельская обл., учитель **Безумова В.А.**;

— Габлевский Эдуард и Савинов Сергей, г. Белово Кемеровской обл., поселок Краснобродский, школа № 31, учитель **Зайцева Л.А.**;

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

— Кондаков Роман, г. Ярославль, школа № 33, учитель **Ярцева О.В.**;

— Маковский Александр и Слимов Никита, Москва, школа № 1399, учитель **Шнейдер Е.В.**;

— Приказчикова Антонина, Сафарова Алина и Сергеев Андрей, средняя школа села Ириновка, Новобурасский р-н Саратовской обл., учитель **Брунов А.С.**;

— Тымченко Александр, Приморский край, г. Владивосток, школа № 22, учитель **Гниляк Т.В.**;

— Хасаншин Айдар, Республика Татарстан, г. Бавлы, гимназия № 4, учитель **Шафиков Н.Р.**;

— Ядзевичюс Стасис, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**

### 3. Еще одна задача “Прогноз погоды”

*Ответ.* Достоверный прогноз: ясно, слабый ветер, гололед.

Правильные ответы представили:

— Акбарова Снежанна, Габдуллин Руслан, Зайнакаев Вадим, Коробейникова Полина, Матиенко Анна, Султанов Алмаз и Швецова Анна, Республика Башкортостан, Краснокамский р-н, село Николо-Березовка, школа № 1, учитель **Ситдикова А.Г.**;

— Вылка Татьяна, основная школа поселка Каратайка, Архангельская обл., учитель **Безумова В.А.**;

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

— Мнацаканян Ашот, средняя школа поселка Новопетровский Московской обл., учитель **Артамонова В.В.**;

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

— Ядзевичюс Стасис, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**

Программы решения задач, предложенных для самостоятельной работы в статье “Решение линейного дифференциального уравнения без лишних операций”, прислали:

— Акбарова Снежанна и Габдуллин Руслан, Республика Башкортостан, Краснокамский р-н, село Николо-Березовка, школа № 1, учитель **Ситдикова А.Г.**;

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

— Леопольд Елена, рабочий поселок Яя Кемеровской обл., школа № 2, учитель **Щербакова О.Б.**;

— Яновский Виталий, Москва, гимназия № 1530, учитель **Шамшев М.В.**

Елена Леопольд, Снежанна Акбарова и Руслан Габдуллин, как приславшие лучшие ответы (они выполнили по 4 задания), будут награждены дипломами. Поздравляем!

Правильное решение числового ребуса “ИКС в квадрате” представили также:

— Тимофеева Юлия, Республика Башкортостан, г. Стерлитамак, школа № 1, учитель **Орлова Е.В.**;

— Ядзевичюс Стасис, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Каликина Т.В.**

Стасис Ядзевичюс прислал также правильные решения задач “В небольшом городке”, “Иностранные языки”, “«Классные» белые грибы”, числовых ребусов “Сколько будет ОГОГО + УГУГУ?”, “ИКС в квадрате”, “Чему равны БИТ и БАЙТ?” и головоломки “Получить из «ПРИКАЗ» «КАПРИЗ»”.

Правильные ответы на задание “Восемь вопросов” представила также Галушкова Карина, Республика Карелия, поселок Надвоицы, школа № 1, учитель **Богданова Л.М.**

Спасибо всем!

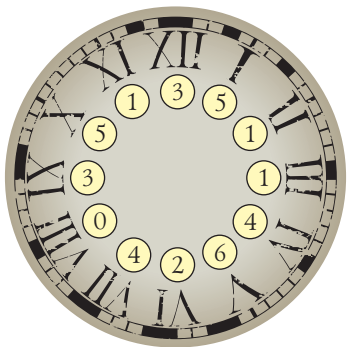
### Календарь из часов

Давайте сделаем часы, которые являются календарем (да-да — именно так!).

Нарисуем на бумаге круг и впишем в него числа 1, 2, 3, ..., 12, как на циферблате часов (см. с. 27).

Рядом с каждым числом (ближе к центру) запишем числа в кружочках так, как показано на рисунке. Все — календарь готов! С его помощью можно решать разные задачи. Рассмотрим их.

Будем считать, что числа на циферблате — это номера месяцев (1 — январь, 2 — февраль, ..., 12 — декабрь), а номера в кружочках — порядковые номера дней недели.



День недели	Номер
Понедельник	1
Вторник	2
...	
Суббота	6
Воскресенье	0 (условно)

### Задача 1

Требуется узнать, каким днем недели будет 26 октября 2011 года.

*Решение*

Октябрь — 10-й месяц. Против числа 10 стоит 5. Прибавляем к дате (26) число в кружочке (5):  $26 + 5 = 31$ . Определяем остаток от деления 31 на 7 — получим 3. Это и есть порядковый номер дня недели. Итак, 26 октября 2011 года будет среда.

### Задача 2

Вася родился 15 мая. В какой день недели он будет праздновать свой день рождения в 2011 году?

*Решение*

Май — 5-й месяц. Против числа 5 стоит 6. Прибавляем к дате (15) число в кружочке (6):  $15 + 6 = 21$ . В остатке от деления 11 на 7 получим 0. Остаток и есть порядковый номер дня недели. Значит, день рождения Васи будет в воскресенье.

### Задача 3

Какие числа соответствуют вторникам в августе 2011 года?

*Решение*

К числам 0, 7, 14, 21 и 28 прибавляем порядковый номер дня недели (2), а затем вычитаем число, стоящее в кружочке рядом с месяцем (для августа — 0):

$$0 + 2 - 0 = 2$$

$$7 + 2 - 0 = 9$$

$$14 + 2 - 0 = 16$$

$$21 + 2 - 0 = 23$$

$$28 + 2 - 0 = 30$$

Следовательно, вторники приходятся на 2, 9, 16, 23 и 30 августа.

### Задача 4

Занятия в кружке по программированию в декабре 2011 года назначены на среды. Определить числа, в которые будут проходить занятия кружка.

*Решение*

К числам 0, 7, 14, 21 и 28 прибавляем порядковый номер дня недели (3), а затем вычитаем число, стоящее в кружочке рядом с месяцем (для декабря — 3):

$$0 + 3 - 3 = 0$$

$$7 + 3 - 3 = 7$$

$$14 + 3 - 3 = 14$$

$$21 + 3 - 3 = 21$$

$$28 + 3 - 3 = 28$$

Итак, занятия состоятся 7, 14, 21 и 28 декабря.

## Задания для самостоятельной работы

- Подготовив описанный календарь, определите:
    - в какой день недели будет день вашего рождения в 2011 году;
    - какие числа соответствуют субботам в августе 2011 года.
  - Подумайте, как составлен описанный календарь. Будет ли он действовать в другие года, например, в 2012-м? Можно ли на основе календаря 2011 года составить календарь на следующий год?
  - Разработайте программу (на изучаемом языке программирования), с помощью которой можно решать задачи двух описанных типов.
  - Подготовьте лист электронной таблицы Microsoft Excel, с использованием которого можно решать задачи, аналогичные задачам 1 и 2 в статье.
- Ответы по п. 2 и/или программы и/или лист Excel, пожалуйста, присылайте в редакцию. Фамилии всех приславших будут опубликованы, а лучшие ответы мы поощрим.

## Литература

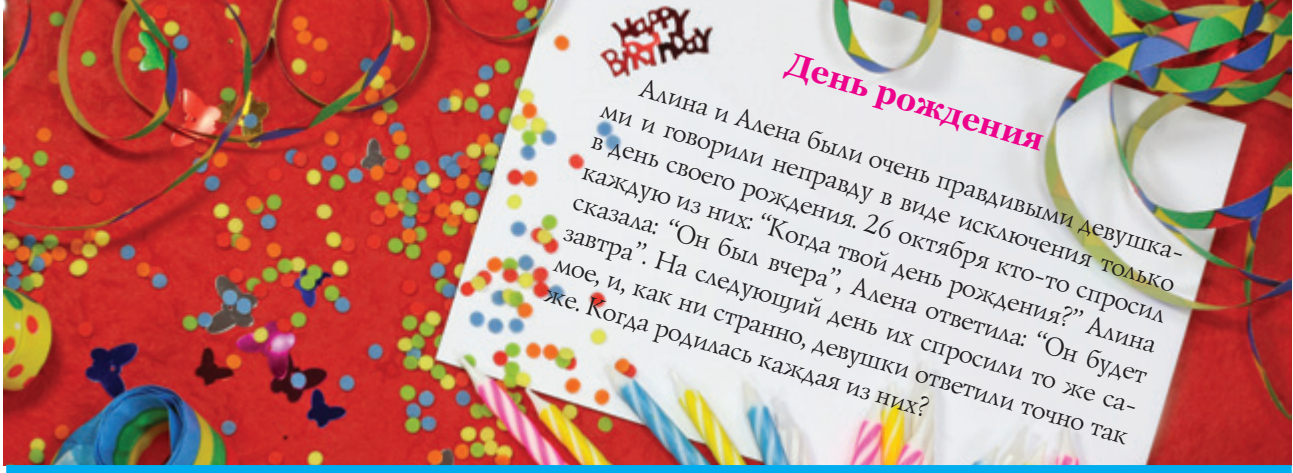
- Розтуллер В.М. Часы-календарь. / Квант, 1973, № 12.

## В одном купе

Как-то раз судьба свела в одном купе поезда известного историка, поэта, прозаика и драматурга. Это были Антонов, Борисенко, Симонян и Дмитриев. Когда поезд тронулся, они углубились в чтение. Оказалось, что каждый из них взял с собой книгу, написанную одним из пассажиров купе. Антонов и Борисенко, дочитав каждый свою книгу, условились на завтра обменяться ими. На следующий день поэт читал пьесу. Прозаик, очень молодой человек, выпустивший свою первую книгу, говорил, что он в жизни не читал и не читает ничего по истории. Историк Борисенко читал произведение Дмитриева. И, конечно, никто из пассажиров не читал книгу, написанную им самим.

Что читал каждый из пассажиров? Каковы профессии каждого из спутников?

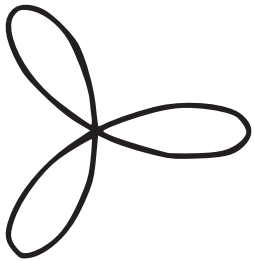




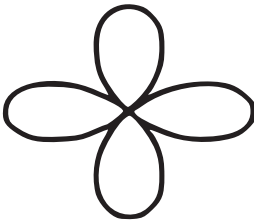
## СЕМИНАР

### Трилистник и другие

Мы продолжаем (см. [1–2]) рассматривать изображения так называемых “замечательных кривых”. В этот раз — кривых, вид которых показан на *рис. 1* и *2*.



*Рис. 1*



*Рис. 2*

На *рис. 1* изображена кривая, которую называют “трилистник”.

В статьях [1–2] для построения линий были использованы их параметрические уравнения. Здесь мы также сделаем это, но сначала приведем уравнение трилистника в полярных координатах:

$$\rho = a \cos 3\varphi \quad (1)$$

В полярных координатах положение точки на плоскости определяется расстоянием  $\rho$  от начала координат и углом  $\varphi$  между радиус-вектором точки и лучом, проходящим через начало координат (он, как правило, совпадает с осью  $x$  в прямоугольной системе координат).

Уравнение (1) определяет зависимость  $\rho$  от угла  $\varphi$ .

Из уравнения в полярных координатах можно легко получить параметрические уравнения трилистника:

$$\begin{aligned} x &= a \cos 3\varphi \cos \varphi; \\ y &= a \cos 3\varphi \sin \varphi. \end{aligned}$$

Теперь, используя параметрические уравнения линии, можно получить ее изображение в программе — для этого надо рассчитать значения координат  $x$  и  $y$  для всех углов от 1 до 360 градусов через

1 градус и поставить точку в соответствующем месте экрана. Программа на школьном алгоритмическом языке, решающая такую задачу для трилистника, имеет вид:

**алг** Трилистник

**нач** цел  $x, y, \text{угол}, x_0, y_0, a$

$a := 100$

| Устанавливаем графический режим

| работы экрана

видео(17)

| Координаты центра экрана

$x_0 := \text{цел}(\text{макс}X/2); y_0 := \text{цел}(\text{макс}Y/2)$

| Для всех целых значений углов

**нц** для угол от 1 до 360

| Переводим угол в радианы

$\text{угол}2 := 6.28 * \text{угол}/360$

| Рассчитываем координаты

| соответствующей точки кривой

$x := x_0 + \text{цел}(a * \cos(3 * \text{угол}2) * \cos(\text{угол}2))$

$y := y_0 + \text{цел}(a * \cos(3 * \text{угол}2) * \sin(\text{угол}2))$

| и изображаем эту точку

точка( $x, y$ )

**кц**

**кон**

#### Примечания

1.  $x_0$  и  $y_0$  — координаты центра экрана (с учетом этих координат рассчитываются значения  $x$  и  $y$ ); значения  $x_0$  и  $y_0$  зависят от величин  $\text{макс}X$  и  $\text{макс}Y$ , равных, соответственно, максимальному значению координат  $x$  и  $y$  в выбранном режиме работы экрана.

2.  $\text{угол}2$  — величина угла в радианах.

3. Функция **цел** возвращает целую часть ее вещественного аргумента.

Можно также использовать электронную таблицу Microsoft Excel (необходимые формулы запишите самостоятельно):

	A	B	C	D
1	Угол в градусах	Угол в радианах	$x$	$y$
2	0	0	100	0
3	5	0.087	96.23	8.41
...	...	...	...	...
73	355	6.192778	95.95094	-8.6984

По расчетным данным можно построить график, изображающий “нашу” кривую (в Мастере диаграмм тип — Точечная).

О том, как называется линия, представленная на *рис. 2*, вы конечно же уже поняли...

## Задания для самостоятельной работы

1. Используя электронную таблицу Microsoft Excel или/и разработав программу (на языке программирования, который вы изучаете), получите изображение линии на рис. 2. Ее уравнение в полярных координатах:

$$\rho = a \cos 2\varphi \quad (2)$$

2. Определите, как будет выглядеть изображение, если использовать такие уравнения линии в полярных координатах:

а)  $\rho = a \cos 6\varphi$ ;

б)  $\rho = a \cos 7\varphi$ .

3. Установите, что и как определяет значение  $k$  в уравнении линии в полярных координатах  $\rho = a \cos k\varphi$ .

Результаты, пожалуйста, присылайте в редакцию. Фамилии всех приславших ответы будут опубликованы, а лучшие ответы мы поощрим.

## Литература

1. Как выглядит кардиоида? / “В мир информатики” № 119 (“Информатика” № 1/2009).

2. Еще две замечательные кривые. / “В мир информатики” № 124 (“Информатика” № 6/2009).

## Считающие чертежи

Что такое “парабола”, конечно, знают многие читатели. На рис. 3 изображена именно она — график функции  $y = x^2$ . Правда, построен он средствами программы Microsoft Excel, поэтому пропорции между значениями по осям  $x$  и  $y$  не соблюдены, но для нас в данном случае это не принципиально. Кроме того, на рис. 3 есть “лишние” числа и линии. Вот о них и поговорим подробнее. Числа на параболе соответствуют абсолютным значениям координаты  $x$  точек, обозначенных кружочками. То, что зеленая линия, соединяющая точки 8 и 8, на оси ординат будет проходить через число 64, факт очевидный — это следует из определения параболы. А через какую точку проходит красная линия, соединяющая точки 8 и 6? Оказывается — через точку 48! (Последний символ — это не знак факториала, а знак удивления 😊.) А прямая, связывающая точки 9 и 3, на оси ординат пройдет через точку... — да, конечно, через 27! И, как говорится, “и так далее и тому подобное”. То есть, соединяя прямой (линейкой или

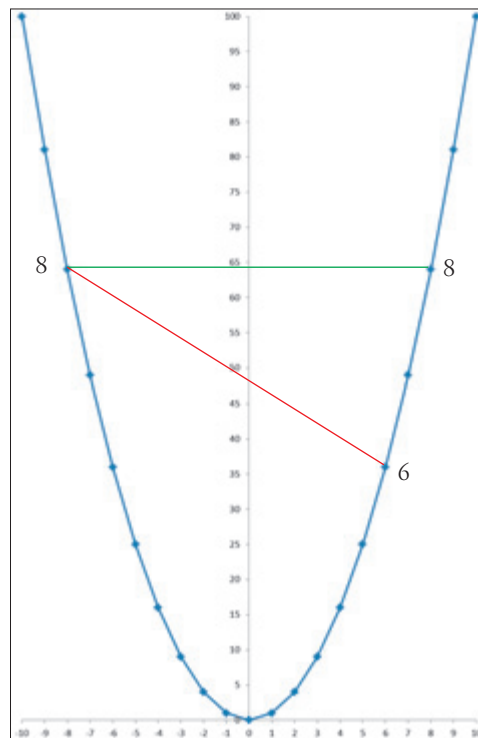


Рис. 3

тонкой ниткой) точки, представленные на правой и левой половинах параболы, можно получать их произведение. Такие “считающие чертежи” называют *номограммами*. Конечно, построенная нами номограмма не имеет большой ценности — таблицу умножения знает каждый. Но номограммы можно строить и для более сложных функций. В таких случаях они значительно сокращают время вычислений.

## Задания для самостоятельной работы

1. Используя электронную таблицу Microsoft Excel, получите параболу, представленную на рис. 3 (без красной и зеленой линий, но со всеми числами).

2. Подумайте, пожалуйста, над вопросом: “Можно ли так же находить (приближенно) произведения и нецелых множителей?”

3. Докажите справедливость того факта, что на оси ординат получается искомое произведение.

Ответы присылайте в редакцию (можно выполнять не все задания).

## ЭТО ПОЛЕЗНО ЗНАТЬ

### Точку в пространстве можно закодировать одним числом!

В XVII веке французский ученый Рене Декарт предложил определять положение объекта в пространстве с помощью системы координат: одной, двух или трех пересекающихся осей. Например, на плоскости точку можно представить парой чисел  $x$  и  $y$ . Число  $x$  задает расстояние по горизонтали от точки до оси абсцисс,  $y$  — расстояние по вертикали до оси ординат.

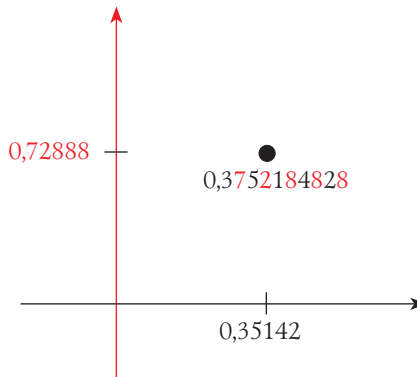
Но, оказывается, существует способ для обозначения точки на плоскости одним числом. Его придумал немецкий математик Георг Кантор. В 1877 году он сделал волнующее открытие, которое огорошило его самого: “Я вижу это, но никак не могу этому поверить!” — написал он своему другу. Что же его так взволновало?

Расположим на плоскости не оси координат, а квадрат со стороной в одну единицу. Любую точку этого квадрата можно обозначить так же, как и в случае с координатными осями на плоскости, то есть с помощью двух чисел, в данном случае — десятичных дробей (ведь сторона квадрата — единица), скажем, 0,35142 и 0,72888.

Так почему бы не соединить попеременно цифры обеих координат и не закодировать эту точку одним числом: 0,3752184828 (см. рисунок справа)? Выходит, наша, равно как и любая другая, точка плоскости будет иметь свое конкретное обозначение, и всякая путаница тут исключается, ведь никакую другую точку тем же числом не обозначить.

Но самое интересное, что тот же принцип может работать и в трехмерном пространстве! Для определения точки в нем даем по очереди цифры десятичных дробей трех координат. Все крайне просто, никаких премудростей!

В заключение — несколько слов об авторе описанной идеи. Георг Кантор родился 3 марта 1845 года в Петербурге. Когда он был еще ребенком, его семья переехала из России в Германию. Именно там он начал изучать математику. Защитив в 1868 г. диссертацию по тео-



рии чисел, он получил степень доктора в Берлинском университете. В 27 лет Кантор опубликовал статью, содержащую общее решение очень сложной математической проблемы — и идеи, выросшие впоследствии в его знаменитую теорию — теорию множеств.

## ПОИСК ИНФОРМАЦИИ

### Шесть вопросов. Вариант 2

Ответы на приведенные ниже вопросы найдите в Интернете или по другим источникам информации.

1. Какая станция Ленинградского (Санкт-Петербургского) метро была открыта 28 декабря 1984 года?

2. Из какого дерева сделаны сваи многих старинных венецианских домов?

3. По русским поверьям, через неделю после Троицы наступало это время. Какое?



4. Личность девушки, послужившей моделью для этой картины, точно не установлена: то ли Анна, то ли Шарлотта Бальгауф (или Бальдуф)... Но достоверно известно, что сама эта картина, написанная в 1745 году, стала одной из самых известных пастелей. Что это за картина? Кто ее автор?

5. По одной из версий, название этой азиатской страны, появившееся в XVI веке, происходит от слова, обозначавшего связку жердей, которые применялись при постройке укреплений. Что это за слово? Каково название страны?

6. Банкноты какой страны считаются самыми древними бумажными деньгами?



*Уважаемые коллеги! Дорогие ребята! Будьте, пожалуйста, внимательными и указывайте все необходимые сведения (имя приводите полностью).*

## “ЛОМАЕМ” ГОЛОВУ

### Как получить три квадрата?

Из спичек составлена фигура в виде ключа.



Переставьте четыре спички так, чтобы получилось три квадрата. В новой фигуре спички не должны находиться на тех местах, где раньше были другие спички.

### Кто выше?



Курсанты военного училища построены в виде каре (прямоугольника). В каждой колонне выбрали самого высокого курсанта, а из них — самого низкого. Им оказался Петр Иванов. Затем в каждой шеренге выбрали самого низкого курсанта, а из них — самого высокого. Им оказался Иван Петров. Кто выше — Иван или Петр?

### Еще один числовой ребус с КУБом

Решите, пожалуйста, числовой ребус:

$$\text{КУБ} = (\text{Б} + \text{У} + \text{К})^3$$

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





# ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ



- ✓ качественная система оценки знаний
- ✓ рекомендации по выполнению работы
- ✓ удобный формат и навигация
- ✓ аттестация по всем темам школьной программы
- ✓ эффективная подготовка к итоговой аттестации и государственной итоговой аттестации

Издательство

**Вако**

**ПРОВЕРИТЬ ЗНАНИЯ ЛЕГКО!**