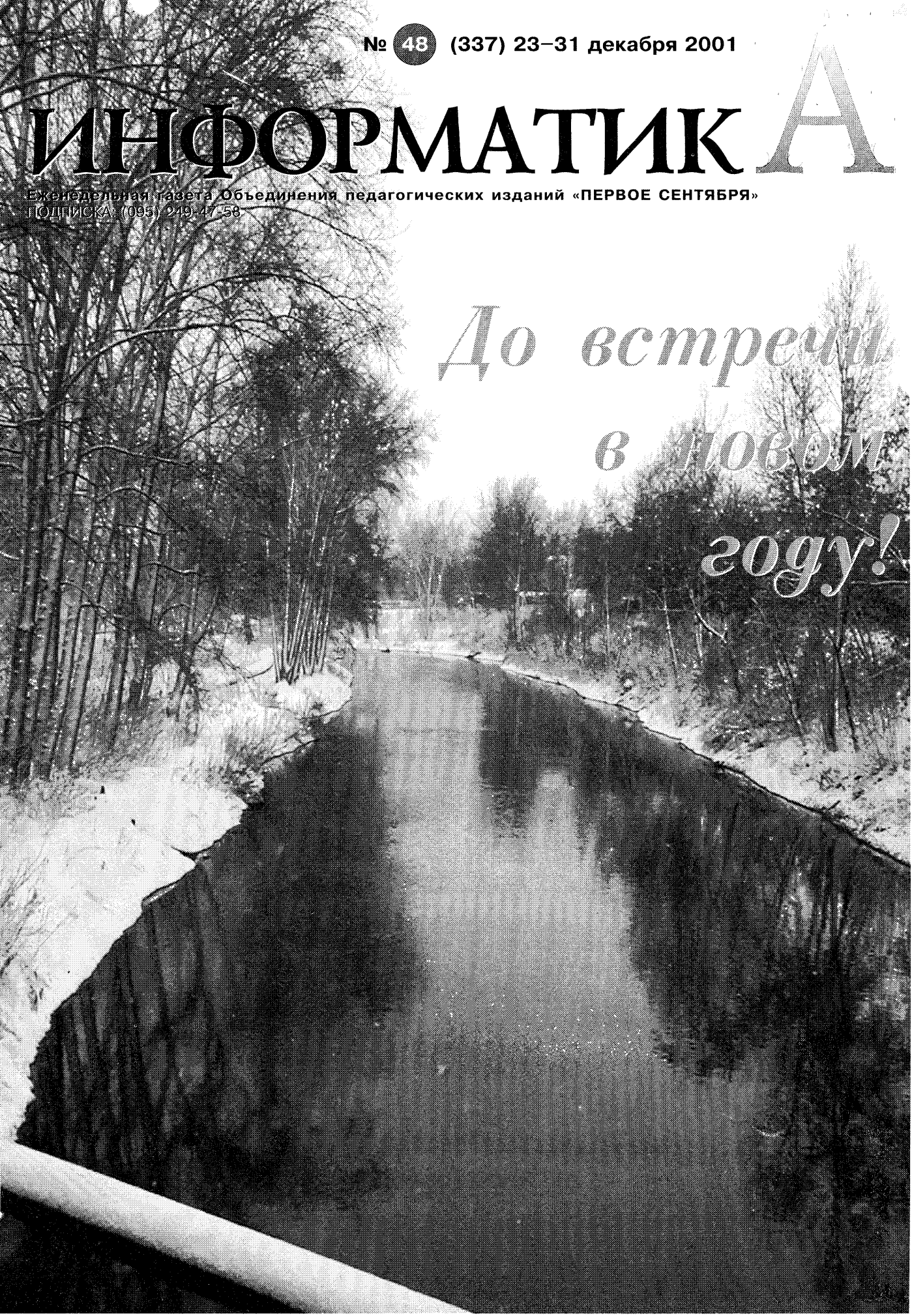


№ 48 (337) 23–31 декабря 2001

ИНФОРМАТИК А

Еженедельная газета Объединения педагогических изданий «ПЕРВОЕ СЕНТЯБРЯ»
ПОДПИСКА: (095) 249-47-58

*До встречи
в новом
году!*



Олимпиады по информатике.

Пути к вершине

Лекции читает Е.В. Андреева

Лекция 6. Комбинаторные задачи

Задачи дискретной математики, к которым относится большинство олимпиадных задач по информатике, часто сводятся к перебору различных комбинаторных конфигураций объектов и выбору среди них наилучшего, с точки зрения условия той или иной задачи. Поэтому знание алгоритмов генерации наиболее распространенных комбинаторных конфигураций является необходимым условием успешного решения олимпиадных задач в целом. Важно также знать количество различных вариантов для каждого типа комбинаторных конфигураций, так как это позволяет реально оценить вычислительную трудоемкость выбранного алгоритма решения той или иной задачи на перебор вариантов и, соответственно, его приемлемость для решения рассматриваемой задачи, с учетом ее размерности. Кроме того, при решении задач полезным оказывается умение для каждой из комбинаторных конфигураций выполнять следующие операции: по имеющейся конфигурации получать следующую за ней в лексикографическом порядке; определять номер данной конфигурации в лексикографической нумерации всех конфигураций; и, наоборот, по порядковому номеру выписывать соответствующую ему конфигурацию.

Перечисленные подзадачи в программировании обычно рассматривают для следующих комбинаторных конфигураций: перестановки элементов множества, подмножества множества, сочетания из n элементов множества по k элементов (k -элементные подмножества множества, состоящего из $n \geq k$ элементов), размещения (упорядоченные подмножества множества, то есть отличающиеся не только составом элементов, но и порядком элементов в них), разбиения множества (множество разбивается на подмножества произвольного размера так, что каждый элемент исходного множества содержится ровно в одном подмножестве), разбиения натуральных чисел на слагаемые, правильные скобочные последовательности (различные правильные взаимные расположения n пар открывающихся и закрывающихся скобок).

Большинство указанных конфигураций были подробно рассмотрены в [1—3]. Однако при генерации различных конфигураций использовались в основном нерекурсивные алгоритмы. Опытные же участники олимпиад в подобных случаях при программировании используют в основном именно рекурсию, с помощью которой решение рассматриваемых задач зачастую

можно записать более кратко и прозрачно. Поэтому для полноты изложения данной темы приведем ряд рекурсивных комбинаторных алгоритмов и рассмотрим особенности применения рекурсии в комбинаторике.

Генерация k -элементных подмножеств

В комбинаторике такие подмножества называют сочетаниями из n элементов по k элементов и обозначают C_n^k . Их количество выражается следующей формулой:

$$C_n^k = \frac{n!}{k!(n-k)!} = \frac{n(n-1)\dots(n-k+1)}{1 \cdot 2 \cdot \dots \cdot k} = \frac{n(n-1)\dots(k+1)}{1 \cdot 2 \cdot \dots \cdot (n-k)}, \quad 0 \leq k \leq n. \quad (1)$$

Однако при программировании гораздо удобнее использовать следующие рекуррентные соотношения:

$$C_n^k = C_{n-1}^{k-1} + C_{n-1}^k, \quad 0 < k < n; \\ C_n^k = \frac{n}{k} C_{n-1}^{k-1}, \quad 0 < k \leq n. \quad (2)$$

Объясняется это тем, что в формуле (1) числитель и знаменатель растут очень быстро, поэтому в силу особенностей компьютерной арифметики не всегда возможно точно вычислить значение C_n^k , даже когда последнее не превосходит максимально представимое целое число.

При фиксированном значении n максимального значения число сочетаний достигает при $k = n/2$ (вернее, для четного n максимум один, и он указан, а для нечетного — максимум достигается на двух соседних значениях k : $[n/2]$ и $[n/2] + 1$). Поэтому особенно полезной оказывается следующая оценка для четных n [4] (очевидно, что при нечетных n отличия будут минимальными), основанная на формуле Стирлинга:

$$C_n^{n/2} = \frac{2^n}{\sqrt{\pi n/2}} (1 + O(1/n)). \quad (3)$$

Если допустить, что за время, ответное для решения задачи, мы можем перебрать около 10^6 вариантов, то из формулы (3) следует, что генерацию всех сочетаний из n элементов для любого фиксированного k можно проводить для $n \leq 24$.

План публикаций лекций курса "Олимпиады по информатике. Пути к вершине" на "Страницах повышения квалификации".

Номер лекции	Номер газеты
1	38/2001
2	40/2001
3	42/2001
4	44/2001
5	46/2001
6	48/2001
7	6/2002
8	8/2002
9	10/2002
10	12/2002
11	14/2002
12	16/2002

Уважаемые читатели, пожалуйста, обратите внимание на то, что лекции 7—12 будут опубликованы в первом полугодии 2002 г.

Обычно генерацию всех k -элементных подмножеств проводят в лексикографическом порядке, тем более что в данном случае это не приводит ни к усложнению алгоритма, ни к увеличению его вычислительной трудоемкости. Напомним, что порядок подмножеств называется лексикографическим, если для любых двух подмножеств справедливо, что раньше должно быть сгенерировано то из них, из индексов элементов которого можно составить меньшее k -значное число в n -ричной системе счисления (или в десятичной, для $n < 10$). Так, для $n = 6$ и $k = 3$ сочетание из третьего, первого и пятого элементов должно быть сгенерировано раньше, чем из второго, третьего и четвертого, так как $135 < 234$.

Рассмотрим рекурсивный алгоритм решения данной задачи. Идея сведения данной задачи к задаче меньшей размерности следующая. Первым элементом подмножества может быть любой элемент, начиная с первого и заканчивая $(n - k + 1)$ -м элементом. После того, как индекс первого элемента подмножества зафиксирован, осталось выбрать $k - 1$ элемент из элементов с индексами большими, чем у первого. Далее поступаем аналогично. Когда выбран последний элемент, то мы достигли конечного уровня рекурсии и выбранное подмножество можно обработать (проанализировать или распечатать). В предлагаемой ниже программе массив a содержит значения элементов исходного множества и может быть заполнен произвольным образом. В массиве p будем формировать очередное сочетание из k элементов.

```
const nmax = 24;
type list = array[1..nmax] of integer;
var k, i, j, n, q: integer;
    a, p: list;
procedure print(k: integer);
var i: integer;
begin
    for j := 1 to k do
        write(p[j]:4);
    writeln
end; {print}

procedure cnk(n, k: integer);

procedure gen(m, L: integer);
var i: integer;
begin
    if m = 0 then print(k) else
        for i := L to n - m + 1 do
            begin
                p[k - m + 1] := a[i];
                gen(m - 1, i + 1)
            end
        end; {gen}
end; {cnk}

begin {cnk}
    gen(k, 1)
end; {cnk}

begin {main}
    readln(n, k);
    for i := 1 to n do
        a[i] := i;
        {заполнить массив можно и по-другому}
    cnk(n, k)
end.
```

Заметим, что собственно генерация сочетаний производится в рекурсивной подпрограмме `gen`. Она имеет следующие параметры: m — сколько элементов из множества нам еще осталось выбрать и L — начиная с какого элемента исходного множества следует выбирать эти m элементов. Обратите внимание, что именно вложенная структура описания процедур `cnk` и `gen` позволяет не передавать при рекурсивных вызовах значения n и k , а из основной программы обращаться к процедуре `cnk` с параметрами, соответствующими постановке задачи, не вдаваясь в подробности ее решения. Такой способ будем применять и в дальнейшем.

Генерация всех подмножеств данного множества

При решении олимпиадных задач чаще всего заранее неизвестно, сколько именно элементов исходного множества должно входить в искомое подмножество, то есть необходим перебор всех подмножеств. Однако если требуется найти минимальное подмножество, то есть состоящее как можно из меньшего числа элементов (или максимальное подмножество), то эффективнее всего организовать перебор так, чтобы сначала проверялись все подмножества, состоящие из одного элемента, затем из двух, трех и т.д. элементов (для максимального подмножества — в обратном порядке). В этом случае первое же подмножество, удовлетворяющее условию задачи, и будет искомым и дальнейший перебор следует прекратить. Для реализации такого перебора можно воспользоваться, например, процедурой `cnk`, описанной в предыдущем разделе. Введем в нее еще один параметр: логическую переменную `flag`, которая будет обозначать, удовлетворяет текущее сочетание элементов условию задачи или нет. При получении очередного сочетания вместо его печати обратимся к процедуре его проверки `check`, которая и будет определять значение флага. Тогда начало процедуры `gen` следует переписать так:

```
procedure gen(m, L: integer);
var i: integer;
begin
    if flag then exit;
    if m = 0 then
        begin
            check(p, k, flag);
        end
    else ...
```

Далее процедура дословно совпадает с предыдущей версией. В основной же программе единственное обращение к данной процедуре следует заменить следующим фрагментом:

```
k := 0;
flag := false;
repeat
    k := k + 1;
    cnk(n, 1, flag)
until flag or (k = n);
if flag then print(k)
else writeln('no solution');
```

Очевидно также, что в основной программе запрос значения переменной k теперь не производится.

Существует также альтернативный подход к перебору всех подмножеств того или иного множества. Каждое подмножество можно охарактеризовать, указав относительно каждого элемента исходного множества, принадлежит оно данному подмножеству или нет. Сделать это можно, поставив в соответствие каждому элементу множества 0 или 1. То есть каждому подмножеству соответствует n -значное число в двоичной системе счисления (строго говоря, так как числа могут начинаться с произвольного количества нулей, которые значащими цифрами не считаются, то следует заметить, что в соответствие ставятся n - или менее значные числа). Отсюда следует, что полный перебор всех подмножеств данного множества соответствует перебору всех чисел в двоичной системе счисления от

$$\underbrace{0\dots01}_{n-1} \text{ до } \underbrace{1\dots1}_n.$$

Теперь легко подсчитать и количество различных подмножеств данного множества. Оно равно $2^n - 1$ (или 2^n , с учетом пустого множества). Таким образом, сопоставляя два способа перебора всех подмножеств данного множества, мы получили следующую формулу:

$$C_n^0 + C_n^1 + \dots + C_n^n = 2^n. \quad (4)$$

То есть в рамках сделанной выше оценки на количество допустимых вариантов в переборе мы можем рассмотреть все подмножества исходного множества только при $n \leq 20$.

Прежде чем перейти к рассмотрению программ, соответствующих второму способу перебора, укажем, когда применение этих программ целесообразно. Во-первых, данные программы легко использовать, когда необходимо в любом случае перебрать все подмножества данного множества (например, требуется найти все решения, удовлетворяющие тому или иному условию). Во-вторых, когда с точки зрения условия задачи не имеет значения, сколько именно элементов должно входить в искомое подмножество. На примере такой задачи мы и напишем программу генерации всех подмножеств исходного множества в лексикографическом порядке. Задача взята из книги [5].

Условие. Даны целочисленный массив $a[1..N]$ ($N \leq 20$) и число M . Найти подмножество элементов массива $a[i_1], a[i_2], \dots, a[i_k]$ такое, что $1 \leq i_1 < i_2 < i_3 < \dots < i_k \leq N$ и $a[i_1] + a[i_2] + \dots + a[i_k] = M$.

Решение. В качестве решения приведем процедуру генерации всех подмножеств, которые можно составить из элементов массива, и функцию проверки конкретного подмножества на соответствие условию задачи.

```
function check(j: longint): boolean;
var k: integer; s: longint;
begin
  s := 0;
  for k := 1 to n do
    {данное условие означает, что в k-й справа
     позиции числа j, во 2-й системе, стоит 1}
    if ((j shr (k - 1)) and 1) = 1
    then s := s + a[k];
```

```
if s = m then
begin
  for k := 1 to n do
    if ((j shr (k - 1)) and 1) = 1
    then write(a[k]:4);
  writeln
end
end;
```

```
procedure subsets(n: integer);
var q, j: longint;
begin
  {помещаем в q число 2^n}
  q := 1 shl n;
  {цикл по всем подмножествам}
  for j := 1 to q - 1 do
    if check(j) then exit
  end;
```

Заметим, что если все элементы в массиве положительные, то, изменив порядок рассмотрения подмножеств, решение приведенной выше задачи можно сделать более эффективным. Так, если сумма элементов какого-либо подмножества уже больше, чем M , то рассматривать подмножества, включающие его в себя, уже не имеет смысла. Пересчет же сумм можно оптимизировать, если каждое следующее сгенерированное подмножество будет отличаться от предыдущего не более чем на один элемент (такой способ перечисления подмножеств показан в [2]). Приведенная же программа чрезвычайно проста, но обладает одним недостатком: мы не можем ни в каком случае с ее помощью перебрать все подмножества множества, состоящих из более чем 30 элементов, что обусловлено максимальным числом битов, отводимых на представление целых чисел в Турбо Паскале (32 бита). Но, как уже было сказано выше, на самом деле перебор всех подмножеств у множеств большей размерности вряд ли возможен за время, отведенное для решения той или иной задачи.

Генерация всех перестановок n -элементного множества

Количество различных перестановок множества, состоящего из n элементов, равно $n!$. В этом нетрудно убедиться: на первом месте в перестановке может стоять любой из n элементов множества, после того как мы на первом месте зафиксировали какой-либо элемент, на втором месте может стоять любой из $n - 1$ оставшегося элемента и т.д. Таким образом, общее количество вариантов равно $n(n - 1)(n - 2) \dots 3 \cdot 2 \cdot 1 = n!$. То есть рассматривать абсолютно все перестановки мы можем только у множеств, состоящих из не более чем 10 элементов.

Рассмотрим рекурсивный алгоритм, реализующий генерацию всех перестановок в лексикографическом порядке. Такой порядок зачастую наиболее удобен при решении олимпиадных задач, так как упрощает применение метода ветвей и границ, который будет описан ниже. Обозначим массив индексов элементов — p . Первоначально он заполнен числами 1, 2, ..., n , которые в дальнейшем будут меняться местами. Параметром i рекурсивной процедуры Perm служит место в массиве p , начиная с которого должны быть получены все перестановки

правой части этого массива. Идея рекурсии в данном случае следующая: на i -м месте должны побывать все элементы массива p с i -го по n -й, и для каждого из этих элементов должны быть получены все перестановки остальных элементов, начиная с $(i + 1)$ -го места, в лексикографическом порядке. После получения последней из таких перестановок исходный порядок элементов должен быть восстановлен.

{описание переменных совпадает с приведенным выше}
procedure Permutations(n : **integer**);

```
procedure Perm( $i$ : integer);
var  $j, k$ : integer;
begin
  if  $i = n$  then
    begin for  $j := 1$  to  $n$  do  $\text{write}(a[p[j]], ' ');$ 
     $\text{writeln}$  end
  else
    begin
      for  $j := i + 1$  to  $n$  do
        begin
          Perm( $i + 1$ );
           $k := p[i]; p[i] := p[j]; p[j] := k$ 
        end;
        Perm( $i + 1$ );
      {циклический сдвиг элементов  $i..n$  влево}
       $k := p[i];$ 
      for  $j := i$  to  $n - 1$  do  $p[j] := p[j + 1];$ 
       $p[n] := k$ 
    end
  end; {Perm}
begin {Permutations}
  Perm(1)
end;
begin {Main}
   $\text{readln}(n);$ 
  for  $i := 1$  to  $n$  do  $p[i] := i;$ 
  {массив  $a$  может быть заполнен произвольно}
   $a := p;$ 
  Permutations( $n$ )
end.
```

Заметим, что в данной программе массив p можно было и не использовать, а переставлять непосредственно элементы массива a .

Разбиения множества

Число разбиений n -элементного множества на k блоков произвольного размера, но таких, что каждый элемент множества оказывается "приписан" к одному из блоков, выражается числом Стирлинга второго рода $S(n, k)$ [6, 7]. Очевидно, что $S(n, k) = 0$ для $k > n$. Если согласиться, что существует только один способ разбиения пустого множества на нулевое число непустых частей, то $S(0, 0) = 1$ (именно такая договоренность, как и в случае с факториалом, приводит в дальнейшем к универсальным формулам). Так как при разбиении непустого множества нужна по крайней мере одна часть, $S(n, 0) = 0$ при $n > 0$. Отдельно интересно также рассмотреть случай $k = 2$. Если непустое множество разделили на две непустые части, то в первой части может оказаться любое подмножество исходного множества, за исключением подмножеств, включающих в себя последний элемент множества, а оставшиеся элементы

автоматически попадают во вторую часть. Такие подмножества можно выбрать $2^{n-1} - 1$ способами, что и соответствует $S(n, 2)$ при $n > 0$.

Для произвольного k можно рассуждать так. Последний элемент либо будет представлять собой отдельный блок в разбиении, и тогда оставшиеся элементы можно разбить уже на $k - 1$ частей $S(n - 1, k - 1)$ способами, либо помещаем его в непустой блок. В последнем случае имеется $kS(n - 1, k)$ возможных вариантов, поскольку последний элемент мы можем добавлять в каждый блок разбиения первых $n - 1$ элементов на k частей. Таким образом,

$$S(n, k) = S(n - 1, k - 1) + kS(n - 1, k), \quad n > 0. \quad (5)$$

Полезными могут оказаться также формулы, связывающие числа Стирлинга с биномиальными коэффициентами, определяющими число сочетаний:

$$S(n, k) = \sum_{i=k-1}^{n-1} C_{n-1}^i S(i, k - 1); \quad (6)$$

$$S(n, k) = \frac{1}{k!} \sum_{i=1}^k C_k^i i^n (-1)^{k-i}.$$

Если же значение k теперь не фиксировать и рассмотреть все разбиения n -элементного множества, то их количество выражается числом Белла

$$B_n = \sum_{k=0}^n S(n, k); \quad B_{n+1} = \sum_{i=0}^n C_n^i B_i. \quad (7)$$

По формулам (7) можно подсчитать, что в рамках принятых выше допущений можно построить все разбиения множества, состоящего не более чем из 15 элементов ($B_{15} = 1382958545$).

Перейдем теперь к рассмотрению способа генерации всех разбиений исходного множества. Прежде всего следует договориться о том, как обозначать текущее разбиение. Так как в каждом из разбиений участвуют все элементы исходного множества, будем в массиве индексов p записывать, в какой блок попадает каждый из элементов в текущем разбиении. Параметр i в рекурсивной процедуре `part` означает, что на текущем шаге мы именно i -й элемент будем размещать в каждом из допустимых для него блоков, а j как раз и определяет максимальный номер допустимого блока. После того как i -й элемент помещен в один из блоков, рекурсивно решается такая же задача уже для следующего элемента (в данном случае фактически работает универсальная схема перебора с возвратом [8]).

```
procedure partition( $n$ : integer; var  $p$ : list);
procedure part( $i, j$ : integer);
var  $l$ : integer;
begin
  if  $i > n$  then  $\text{print}(n, p)$  else
    for  $l := 1$  to  $j$  do
      begin
         $p[i] := l;$ 
        if  $l = j$  then  $\text{part}(i + 1, j + 1)$ 
        else  $\text{part}(i + 1, j)$ 
      end
    end
end; {part}
```



```
begin {partition}
  part(1,1)
end;
```

Как ни странно, в данном случае процедура print оказывается совсем не тривиальной, если требуется печатать (или анализировать) элементы каждого из блоков разбиения в отдельности. Поэтому приведем возможный вариант ее реализации (как и ранее, распределяли по блокам мы индексы, а печатаем или анализируем сами элементы исходного массива а):

```
procedure print(n: integer; var p: list);
var i, j, imax: integer;
begin
  {определяем количество блоков в разбиении}
  imax := 1;
  for i := 2 to n do
    if p[i] > imax then imax := p[i];
  for i := 1 to imax do {цикл по блокам}
    begin
      for j := 1 to n do
        if p[j] = i then write(a[j]:4);
        write(' ') {блок напечатан}
      end;
      writeln {разбиение напечатано}
    end;
end;
```

Вложенного цикла можно избежать, если требуется, например, подсчитать сумму элементов в каждом из блоков. Тогда, используя дополнительный массив, мы, просматривая элементы массива а последовательно, будем увеличивать значения суммы для блока, соответствующего рассматриваемому элементу (аналогично операции, осуществляемой в алгоритме сортировки подсчетом).

Если при этом рассматривать массив р как n -значное число в n -ричной системе счисления, то можно ввести понятие лексикографического порядка для разбиений множества и ставить задачи определения номера разбиения и обратную ей. Как и ранее (см. [1—3]), они решаются методом динамического программирования и не используют непосредственную генерацию всех разбиений.

Для полноты рассмотрения данной темы самостоятельно измените процедуру partition так, чтобы она генерировала все разбиения, состоящие не более чем из k блоков. После этого напишите процедуру разбиения множества уже на ровно k непустых частей.

Олимпиадные задачи, использующие комбинаторные конфигурации

Пример 1. На одном острове Новой Демократии каждый из его жителей организовал партию, которую сам и возглавил. Отметим, что, ко всеобщему удивлению, даже в самой малочисленной партии оказалось не менее двух человек. К сожалению, финансовые трудности не позволили создать парламент, куда вошли бы, как предполагалось по Конституции острова, президенты всех партий. Посоветовавшись, островитяне решили, что будет достаточно, если в парламенте будет хотя бы один член каждой партии. Помогите островитянам организовать такую, как можно более малочисленный, парламент, в котором будут представлены члены всех партий.

Исходные данные: каждая партия и соответственно ее президент имеют одинаковый порядковый номер от 1 до N ($4 \leq N \leq 150$). Вам даны списки всех N партий

острова Новой Демократии. Выведите предлагаемый вами парламент в виде списка номеров его членов. (Олимпиада стран СНГ, г. Могилев, 1992 г.)

Решение. С теоретической точки зрения данная задача совпадает с задачей генерации всех подмножеств из множества жителей острова Новой Демократии. Причем наиболее подходящим кажется первый подход к решению данной задачи, связанный с генерацией различных сочетаний, начиная с одного жителя. Для полноты изложения этого подхода опишем функцию check, которую следует применить в данной задаче. Исходные данные следует записать в массив s: array[1..150] of set of 1..150, заполнив каждый из n первых элементов этого массива множеством тех партий, в которых состоит тот или иной житель. Тогда функция проверки сочетания из элементов этого массива примет следующий вид:

```
function check(var p: list;
               k: integer): boolean;
var i: integer; ss: set of 1..150;
begin
  ss := [];
  for i := 1 to k do ss := ss + s[p[i]];
  check := (ss = [1..n])
end;
```

Как видно из описания функции, использование типа “множество” позволяет не только упростить данную программу, но и существенно ускорить ее выполнение по сравнению с работой с массивами. Однако большая размерность данной задачи не позволяет считать приведенное решение удовлетворительным во всех случаях. Так, уже для $n = 100$ перебор всех сочетаний из 4 и менее жителей приводит к рассмотрению около 4 миллионов вариантов. Для построения кода, пригодного для решения данной задачи при любых входных данных, несколько изменим описание массива s:

```
s: array[1..150] of
  record name, number: integer;
    partys: set of 1..150
  end;
```

Здесь поле partys совпадает по смыслу с первоначальным описанием массива s, поле name соответствует номеру (то есть фактически имени) жителя, и первоначально данное поле следует заполнить числами от 1 до n согласно индексу элемента в массиве записей, и поле number соответствует количеству элементов во множестве из поля partys, то есть имеет смысл сразу подсчитать, в каком количестве партий состоит тот или иной житель. Теперь следует отсортировать массив s по убыванию значений поля number. Верхнюю оценку для числа членов парламента (k_{max}) подсчитаем, построив приближенное решение данной задачи следующим образом: во-первых, включим в парламент жителя, состоящего в максимальном количестве партий, затем исключим эти партии из остальных множеств и заново найдем в оставшемся массиве элемент с максимальным значением поля number (уже пересчитанного) и включим его в парламент, и так далее, до тех пор, пока сумма значений пересчитанных полей number у жителей, включенных в парламент, не будет равна n . Найденное количество членов парламента и будет k_{max} .

Теперь следует рассматривать сочетания из ($k\text{max} - 1$) элемента, затем из ($k\text{max} - 2$) и т.д. элементов. Если для сочетаний из какого-то рассматриваемого количества элементов k решение найдено не будет, то это означает, что точным является решение с количеством членов парламента $k + 1$. Так как массив s упорядочен, то если решение для того или иного значения k существует, то скорее всего оно будет найдено при рассмотрении в лексикографическом порядке первых же сочетаний по k элементов. Поэтому временная трудоемкость в переборе возникает, только если решения с данным значением k не существует. В такой ситуации можно воспользоваться следующим “ненаучным” приемом: на поиск решения для каждого k , меньшего, чем $k\text{max}$, отведем фиксированное количество времени, скажем, 2—3 секунды (точнее данное время стоит определить экспериментальным путем). Если за отведенное время решение не найдено, то следует считать полный перебор невозможным и закончить выполнение программы. В любом случае мы будем иметь какое-либо решение исходной задачи: точное или приближенное, то есть иногда содержащее больше членов парламента, чем минимально возможно. Однако на практике такой подход почти всегда приводит к точному решению, в силу перебора “с предпочтением”, проводимого для каждого k (невозможность проведения полного перебора для какого-либо k на практике соответствует отсутствию решения для данного k).

Пример 2. Дан автобусный билет с номером, состоящим из N цифр. Расставить между цифрами знаки арифметических операций “+”, “-”, “*”, “/” (целочисленное деление) и скобки таким образом, чтобы значение полученного выражения было равно 100. Можно образовывать многозначные числа из стоящих рядом цифр. Выражение должно быть корректным с точки зрения арифметики. Допустимы лишние скобки, не нарушающие корректности выражения. При вычислениях используется стандартный приоритет операций, число цифр N в номере билета не больше 6. (*V Всероссийская олимпиада по информатике*, г. Троицк, 1993 г.)

Решение. Для построения универсального алгоритма решения данной задачи будем считать слияние двух соседних цифр одной из операций. Тогда между каждой парой соседних цифр может стоять одна из 5 операций. Для N цифр получаем 5^{N-1} различных вариантов расстановки операций. Перебирать все варианты расстановки операций удобнее всего с помощью рассмотрения всех чисел в 5-ричной системе счисления, состоящих не более чем из $N - 1$ цифры, то есть для $N = 6$ от 00000 до 44444. Для перебора таких чисел необходимо написать процедуру прибавления 1 в 5-ричной системе счисления. Для каждого из вариантов расстановки операций перейдем от исходного массива из N цифр билета к массиву из K чисел, где $K = (N - \text{количество операций слияния цифр в рассматриваемом варианте})$. Теперь мы должны рассмотреть все перестановки из $K - 1$ арифметической операции в данном варианте. Каждая перестановка соответствует одному из порядков выполнения арифметических операций. Так, для 4 чисел перестановка номеров операций 3, 1, 2 означает, что сначала нужно выполнить арифметическое действие между 3-м и 4-м числом, затем

между 1-м и 2-м и затем оставшееся. Если результат выполнения действий данного варианта в порядке, соответствующем текущей перестановке, равен искомому числу 100, то задача решена и можно перейти к печати результата. Для данной задачи возможны и более эффективные решения, но в силу ее небольшой размерности комбинаторный перебор оказывается вполне приемлемым.

Пример 3. Губернатор одной из областей заключил с фирмой “HerNet” контракт на подключение всех городов области к компьютерной сети. Сеть создается следующим образом: в области устанавливаются несколько станций спутниковой связи, и затем от каждого города прокладывается кабель до одной из станций. Технология, используемая компанией, требует при увеличении расстояния увеличения толщины кабеля. Таким образом, стоимость кабеля, соединяющего город со станцией, при используемой компанией технологии будет равна kL^2 , где L — расстояние от города до станции, а k — некий коэффициент. Вам требуется написать программу, определяющую минимальные затраты компании на установку сети.

Входные данные. Во входном файле записано число n ($1 \leq n \leq 10$) — количество городов в области. Затем идут положительные вещественные числа: k — коэффициент стоимости кабеля и P — стоимость постройки одной станции. Далее следуют n пар вещественных чисел, задающих координаты городов на плоскости.

Выходные данные. В первую строку выходного файла нужно вывести минимальные затраты на установку сети (с тремя знаками после десятичной точки), во вторую — количество устанавливаемых станций. Далее вывести координаты станций (с тремя знаками после десятичной точки), а затем — список из n целых чисел, в котором i -е число задает номер станции, к которой будет подключен i -й город. (*Кировский командный турнир по программированию*, 2000 г.)

Решение. В силу небольшой размерности мы можем рассмотреть все возможные варианты разбиения городов на группы, подразумевая, что для каждой группы будет установлена своя станция, причем оптимальным образом (найти оптимальное местонахождение станции для одной группы городов можно по формуле, аналогичной формуле нахождения центра масс). Затем нужно из всех разбиений выбрать то, для которого общая сумма затрат на установку сети будет минимальной.

Литература

1. Окулов С.М. Перестановки. “Информатика”, № 7/2000.
2. Окулов С.М. Комбинаторные задачи. “Информатика”, № 10, 13/2000.
3. Усов Б.Б. Комбинаторные задачи. “Информатика”, № 39/2000.
4. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы. Построение и анализ. М.: МЦНМО, 2000.
5. Брудно А.А., Каплан А.И. Московские олимпиады по программированию. М.: Наука, 1990.
6. Кнут Д. Конкретная математика. Основание информатики. М.: “Мир”, 1998.
7. Липский В. Комбинаторика для программистов. М.: “Мир”, 1988.
8. Андреева Е.В. Еще раз о задачах на полный перебор вариантов. “Информатика”, № 45/2000.

Книжный
шкафАЛЕКСЕЙ АНДРЕЕВИЧ
ЛЯПУНОВ

Алексей
Андреевич
Ляпунов /
Редакторы-
составители:
Н.А. Ляпунова,
Я.И. Фет.
Новосибирск:
Филиал "Гео"
Издательства
СО РАН,
Издательство
ИВМиМГ СО
РАН, 2001.

Книга "Алексей Андреевич Ляпунов" из серии "История информатики"

Более двух лет назад, в прошлом веке ☺, мы познакомили наших читателей с материалами уникальной книги "Очерки истории информатики в России" (см. № 29/1999). Осенью 2001 г. в Новосибирске вышла очередная книга из серии "История информатики", посвященная жизни и деятельности выдающегося российского ученого Алексея Андреевича Ляпунова. В истории отечественной науки А.А. Ляпунов занимает особое место в связи с тем, что он сыграл решающую роль в борьбе за признание и развитие кибернетики.

Формально книга состоит из шести следующих разделов:

- I. Об А.А. Ляпунове говорят коллеги, ученики, однополчане, журналисты
- II. А.А. Ляпунов о проблемах образования
- III. А.А. Ляпунов и возрождение советской биологии
- IV. Штрихи к портрету
- V. А.А. Ляпунов поддерживает, рекомендует, критикует...
- VI. Переписка

Но мы не случайно употребили слово "формально". Книгу открывает интереснейшая статья Наталии Алексеевны Ляпуновой "Одиннадцать счастливых лет". В этой статье Наталия Алексеевна — дочь Алексея Андреевича — увлекательно и обстоятельно, приводя множество интересных фактов, рассказывает преимущественно о сибирском периоде жизни А.А. Ляпунова.

Вообще в книге содержится огромное количество интересных и неизвестных ранее широкой публике документов. Нет никакой возможности даже кратко упомянуть все материалы, которые могут представлять интерес для педагогов и учителей информатики, но на один из разделов книги все же хочется обратить особое внимание. Речь идет о статьях, заметках, выступлениях А.А. Ляпунова, посвященных вопросам образования. Вот, например, некоторые фрагменты из статьи "Ученики, отметки, оценки", опубликованной в журнале "Химия и жизнь" в № 1/1969.

Хорошо известно, что разные люди обладают различными наследственными задатками. Эти задатки превращаются в той или иной степени в индивидуальные особенности этих людей в зависимости от тех условий, в которых протекает формирование и развитие личности.

Поэтому благоприятными для развития оказываются, вообще говоря, неодинаковые условия. Встречаются люди, обладающие способностью быстро схватывать и быстро усваивать новые сведения, сопоставлять разные данные и т.д. Есть и "тугодумы" — люди, медленно соображающие, с трудом ориентирующиеся в новом материале и далеко не сразу замечая аналогии.

В то же время эти индивидуальные особенности довольно причудливо переплетаются между собой. Так, например, некоторые легко запоминают новые сведения, но не умеют сопоставлять между собой разные обстоятельства. Другие же интеллектуально медлительны, но способны при наличии достаточного времени охватить весьма обширный материал, глубоко в него вникнуть и сделать далеко идущие заключения. Достаточно вспомнить крупнейших математиков Д.Гильберта и Н.Н. Лузина. Оба они производили при первом знакомстве впечатление туго соображающих и плохо ориентирующихся в новом материале людей. В то же время оба сделали важный вклад в математику и создали выдающиеся научные школы.

<...>

Таким образом, разные личности нуждаются в совершенно разном режиме для своего полноценного индивидуального развития <...>. Одним нужен явно ускоренный режим, другим, наоборот, замедленный. У одних интересы формируются рано, у других — значительно позднее;

у одних интеллектуальная активность пробуждается в молодости, у других — только в зрелые годы.

Поэтому в интересах и личности, и общества в целом нецелесообразно "стричь всех под одну гребенку", предъявлять ко всем одинаковые требования и вести всех по единой педагогической стезе. Напротив, необходим индивидуальный, дифференцированный подход к учащимся.

<...>

При существующей системе, когда классы формируются случайно, без учета индивидуальных особенностей ребят, совершенно нелепо выдвигаемое иногда на местах требование стопроцентной успеваемости <...>. В са-

мом деле, это вынуждает учителя уделять основное внимание отстающим, то есть тем, кто не выдерживает установленного ритма школьных занятий. При этом естественным образом получается, что ребятам, быстро усваивающим материал, учитель уделяет меньше внимания, и они сравнительно скоро приучаются к безделью, тогда как медлительные оказываются перегруженными, а подчас задержанными.

<...> добросовестный учитель, предъявляющий своим ученикам разумные требования, попадает в ряд плохих. Напротив, недобросовестный или малоквалифицированный человек, ставящий отметки безответственно, получает поощрение. Совершенно очевидно, что оценивать работу учителя по выставляемым им же самим оценкам бессмысленно <...>.



В 2002 году "Информатика" предложит подписчикам сотни интересных и полезных материалов. Помимо наших постоянных рубрик (см. с. 10–12, 21–22), мы активно развиваем специальные проекты. По сложившейся традиции в № 8/2002 будет опубликован план тематических выпусков очередного "Жаркого лета" (перечень спецвыпусков прошлых лет см. на с. 23–24). Расширится список курсов, публикуемых на "Страницах повышения квалификации". На с. 13–20 читайте "годовой отчет" редакции "Информатики" перед читателями — тематический обзор и предметно-тематический указатель материалов, опубликованных в 2001 г.

индекс подписки — 32291

в каталоге «Роспечать. Газеты и журналы. 2001» см. с. 88

в каталоге «Роспечать. Газеты и журналы. 2002» см. с. 95

Ф. СП-1

Министерство связи
Российской Федерации
"Роспечать"

АБОНЕМЕНТ на газету

32291

ИНФОРМАТИКА

(индекс издания)

наименование издания

Количество
комплектов

на 20__ год по месяцам

1	2	3	4	5	6	7	8	9	10	11	12

Куда

(почтовый индекс)

(адрес)

Кому

(фамилия, инициалы)

ДОСТАВОЧНАЯ КАРТОЧКА

ПВ

место

ли-
тер

на газету

32291

(индекс издания)

ИНФОРМАТИКА

(наименование издания)

Стои-
мость

подписки
пере-
адресовки

руб.
руб.

Количество
комплек-
тов

на 20__ год по месяцам

1	2	3	4	5	6	7	8	9	10	11	12

Куда

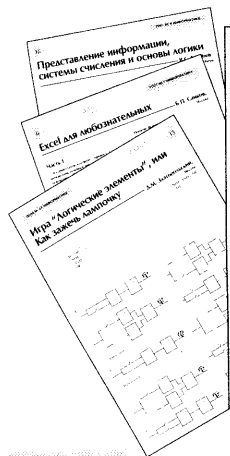
(почтовый индекс)

(адрес)

Кому

(фамилия, инициалы)

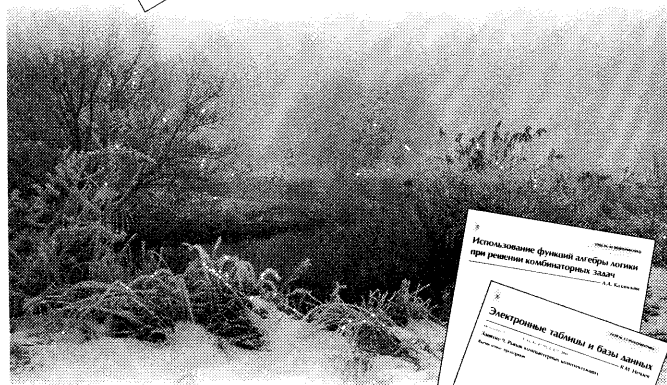




“Уроки”, “Материалы к уроку”

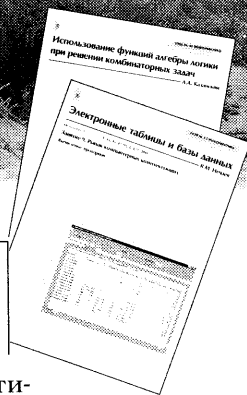
“Уроки”, “Материалы к уроку” – центральные рубрики газеты. Как правило, авторами материалов, публикуемых в этих рубриках, являются наши читатели – учителя, которые делятся методическими разработками и удачными находками с коллегами. Все материалы “урочных” рубрик готовятся таким образом, чтобы любой учитель мог сразу или после несложной адаптации использовать их на собственных уроках. С 1995 года в рубриках “Уроки”, “Материалы к уроку” и “Урок на час” было опубликовано более 200 статей, авторами которых выступили более 100 наших коллег.

Редакция “Информатики” всегда готова рассмотреть предложения от новых авторов.



“Уроки для учителя”

В рубрике “Уроки для учителя” мы публикуем материалы, которые служат повышению квалификации учителей информатики. Все мы по собственному опыту знаем, что нельзя замыкаться исключительно на “школьной информатике”, необходимо стремиться поддерживать свои знания на более высоком уровне, знать больше и глубже, чем это необходимо для проведения обычных уроков. Именно для этого и предназначены материалы этой рубрики. “Страницы повышения квалификации”, которые появятся в “Информатике” в 2001/2002 учебном году, создаются на основе двух рубрик – “Методика” и “Уроки для учителя”.



“Олимпиады”

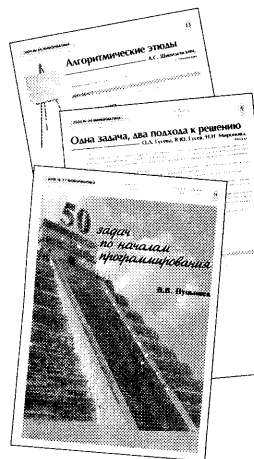
Олимпиадам по информатике разного уровня в нашей газете всегда уделялось большое внимание. На протяжении многих лет мы выпускаем специальные весенние тематические номера, целиком посвященные олимпиадным задачам, а в серии “Жаркое лето-2001” вышел специальный олимпиадный номер, посвященный методике организации, проведения и задачам рыбинских олимпиад.

Но только этими номерами публикации, посвященные олимпиадам, конечно, не ограничиваются. Мы стараемся публиковать материалы олимпиад различного уровня – от школьных до международных. Все последние годы в “Информатике” публикуются разборы задач Всероссийской и Международной олимпиад. Обязательная публикация решений для всех олимпиадных задач – принципиальная позиция редакции. Одна из последних (интереснейших!) публикаций данной рубрики – материалы Пермской олимпиады по базовому курсу информатики.



“Задачи”

В рубрике “Задачи” публикуются разноуровневые задачи по различным темам школьного курса информатики. В течение первых лет выхода газеты в этой рубрике публиковались преимущественно задачи по программированию. Мы и сейчас продолжаем публиковать материалы такого рода. (Например, не так давно закончилась публикация большого сборника задач по программированию, составленного нашим постоянным автором Д.М. Златопольским. Этот сборник осенью 2001 г. выходит в серии книг издательства “Первое сентября” “Я иду на урок.”) Но в последнее время большее внимание уделяется задачам, которые решаются с использованием электронных таблиц, баз данных и даже текстовых и графических редакторов. (В качестве примеров можно привести цикл публикаций “Одна задача, два решения” О.Л. Гусевой, В.Ю. Гусева и Н.Л. Мироновой и статью Г.Г. Мангазеевой “Решаем задачи в текстовом и графическом редакторах.”) Задач такого рода пока не очень много, но потребность в них велика, поэтому мы уделяем большое внимание подготовке соответствующих материалов.



“Круглый стол”

Дискуссионные статьи, опубликованные в знаменитом тематическом выпуске “Школьная информатика в России” (№ 16/1998), стали первыми публикациями в рубрике “Круглый стол” (авторы статей первого “круглого стола”: Е.К. Хеннер, А.И. Сенокосов, А.С. Лесневский, Ю.А. Шафрин, А.Г. Кушниренко). К публикациям в “Круглом столе” редакция относится очень внимательно. С одной стороны, мы стремимся придерживаться принципов свободы слова и плюрализма мнений ☺. С другой стороны, мы заботимся об аргументированности, взвешенности и корректности высказываний. Еще одна особенность материалов данной рубрики (именно поэтому мы и разделили две рубрики — “Круглый стол” и “Мнения”) — их общезначимость, фундаментальность. На материалы “круглого стола” 1998 года ссылаются до сих пор и, надеемся, будут ссылаться еще долго. Не меньшую известность получила статья А.Ю. Уварова “Чему учить на уроках информатики” (№ 1/1999). Большое внимание привлекла и другая публикация этого автора — “Зачем сельской школе Интернет?” (№ 43/2000). Обе статьи ничуть не потеряли актуальности и могут и сегодня служить источником серьезных вопросов для размышления.

“Семинар”

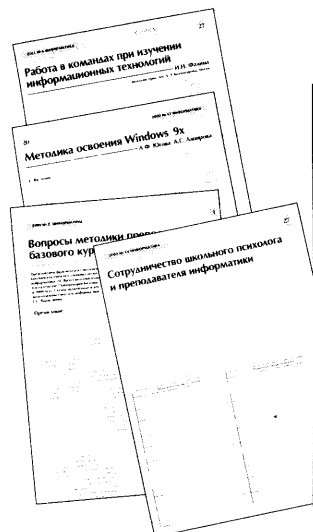
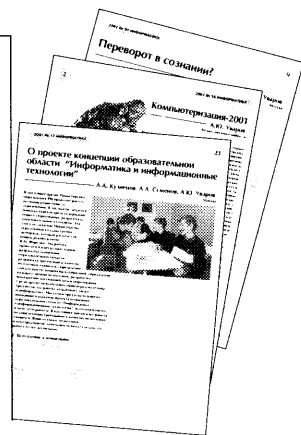
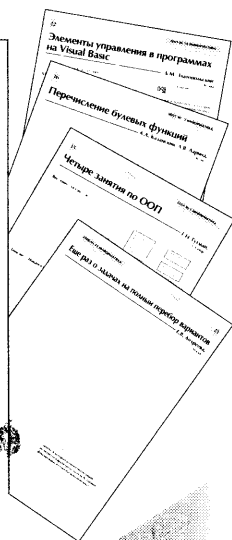
В рубрике “Семинар” публикуются материалы, которые могут использовать учителя, работающие в классах с углубленным изучением информатики, и те, кто занимается с учениками проектной и научно-исследовательской работой. Практически все статьи этой рубрики посвящены интересным, иногда — малоизвестным, иногда — перспективным вопросам информатики. Интересно отметить, что большинство авторов этой рубрики — наши коллеги — учителя, которые делятся интересными находками, уже опробованными ими в собственной практике.

“Мнения”

Рубрика “Мнения” предоставляет возможность каждому, кто связан с информатикой (не обязательно школьной) и “имеет, что сказать” по тому или иному вопросу, высказать свою точку зрения на страницах газеты. В этой рубрике мы предоставляем слово учителям, работникам Министерства образования, авторам учебников и учебных программ. В весенних номерах 2001 г. в рубрике “Мнения” были опубликованы острые дискуссионные материалы, посвященные проекту концепции преподавания информатики в двенадцатилетней школе. Мы не считаем, что на все острые вопросы, которые поднял наш коллега А.Н. Петров в статье по поводу указанной концепции, получены ответы, и намерены вернуться к этой теме.

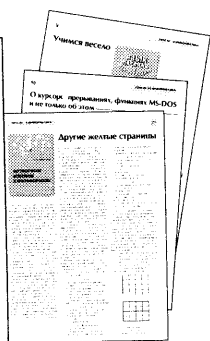
“Методика”

Методические вопросы часто обсуждаются на страницах нашей газеты. Но редакция особенно внимательно, даже придирчиво подходит к методическим материалам. Вы не найдете у нас кабинетных диссертационных исследований, авторы которых стояли у доски лишь во время обязательного “эксперимента”. Публикуемые в “Информатике” методические материалы всегда основаны на реальном (собственном!) опыте работающих учителей. Мы хорошо знаем, что именно методические решения часто способны обеспечить успех учителя, но это должны быть именно решения, а не теоретические исследования, в которых рассматривается “модель учащегося”.



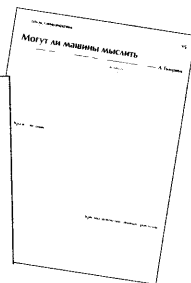
“Внеклассная работа по информатике”

Сразу отметим: далеко не все материалы рубрики “Внеклассная работа” обязательно использовать вне класса. Многие интересные развивающие задания и игры можно с успехом использовать и на уроках. Мало того, их можно использовать и на уроках в старших классах! Опыт показывает, что интересные “развлекательные” задачи увлекают и десятиклассников, и одиннадцатиклассников. Но все же основное назначение этой рубрики — помочь учителю в подборе материалов для проведения недель информатики, викторин, КВНов и других мероприятий, которые, хоть и проводятся обычно в неучебное время после уроков, зачастую требуют от учителя времени на подготовку много больше, чем его требуют обычные уроки. Большое количество материалов рубрики “Внеклассная работа” подготовил наш постоянный автор, московский учитель Д.М. Златопольский. Одна из последних интересных публикаций в этой рубрике — “Интеллектуальная игра”, присланная нашими пермскими коллегами (см. № 36/2001). В этой статье описывается увлекательная игра, которая была проведена с участниками пермской олимпиады по базовому курсу информатики. Мы уверены, что эта публикация будет очень полезна нашим читателям в новом учебном году.



“История информатики”

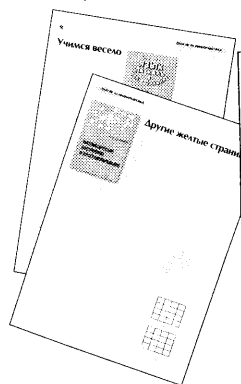
Чтобы интереснейшие публикации, посвященные истории информатики, не “затерялись” среди других материалов, мы выделили их в отдельную рубрику. Отбирая материалы на историческую тему, мы стремимся не ограничиваться “общими местами”, все и так уже хорошо знают этимологию слова “алгоритм” и назначение абака. (Хотя, разумеется, и этим вопросам уделяется должное внимание.) Основное внимание уделяется любопытным, не слишком известным фактам, которые могут служить источником тем для самостоятельных курсовых и исследовательских работ учащихся. Материалы этой рубрики имеют и другую особенность — мы стремимся полнее познакомить наших читателей с историей информатики в России. В этом смысле примечателен тематический выпуск “Очерки истории информатики в России”, который вышел в серии “Жаркое лето-99” (см. № 29/1999). Этот номер был основан на материалах редкой одноименной монографии, которая вышла небольшим тиражом в Новосибирске.



Отметим, что в некоторых случаях название этой рубрики конкретизируется. В этом смысле “История информатики” объединяет и материалы “Истории вычислительной техники”, и материалы “Истории языков программирования”. А вот материалы, посвященные известным ученым-информатикам, всегда выделяются в отдельную рубрику “Информатика в лицах”.

“Книжный шкаф учителя информатики”

Создавая эту рубрику, мы предполагали знакомить наших читателей с интересными и полезными книжными новинками. Но довольно быстро стало понятно, что в наших книжных шкафах стоят не только новинки. Даже напротив — гораздо чаще мы используем в работе замечательные, ставшие классическими книги, вышедшие порой не одно десятилетие назад. А поскольку среди наших подписчиков становится все больше молодых учителей, которые не всегда даже знают о существовании этих книг, мы сочли необходимым уделять им особенное внимание. Рубрика “Книжный шкаф” еще находится в стадии становления. Мы планируем расширить ее, сделать постоянной, налаживаем сотрудничество с книжными издательствами и... внимательно исследуем книжные шкафы наших коллег ☺.



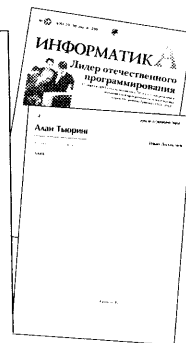
“История науки”

Профессия учителя предполагает широкий кругозор и знания в различных областях, порой не имеющих прямого отношения к специальности. Учителю информатики в этом смысле, пожалуй, едва ли не тяжелее, чем всем остальным. Слишком “много гитик” имеет наша наука. В рубрике “История науки” мы публикуем материалы, в которых рассказывается о научных открытиях, которые являются вехами на пути к современному состоянию наук, имеющих отношение к информатике.



“Информатика в лицах”

А.П. Ершов, А.И. Берг, А.Л. Брудно, А.Тьюринг, Н.Винер, Л.Торвальд — это лишь небольшой (и никак не упорядоченный, фамилии были взяты из случайных номеров газеты) список выдающихся ученых, о которых рассказывалось в нашей газете. Разумеется, некоторым ученым было посвящено несколько статей (и мы намерены и впредь обращаться к этим темам).



Тематический указатель материалов, напечатанных в газете “Информатика” в 2001 году (№ 1–48)

Названия разделов

- А.** Разработки уроков и тем
Материалы для уроков и факультативных занятий
А.1. Как это делаю я
А.2. Семинар
А.3. Урок на час
А.4. Методика
А.5. Внеклассная работа по информатике
Кроссворды, ребусы
А.6. Калейдоскоп
Б. Задачи, задачи...
В. Компьютерные вирусы и защита от них
Г. Начальная школа
Д. Информатика для малышей
Е. Информационные технологии
Е.1. Беседы
Е.2. Узелки на память

- Ж.** Общие вопросы преподавания информатики
Ж.1. Учебники, учебные пособия, словари
З. История науки, история информатики
И. Юбилеи и памятные даты
К. На стенд в кабинете информатики
Л. Предлагаю коллегам
М. Олимпиады, конференции, турниры, школы, курсы, семинары, выставки
Н. Официальные и неофициальные документы
О. Мнения
П. Страницы повышения квалификации
Р. “Книжный шкаф” учителя информатики
С. Читайте в январских номерах
Т. Начните с простого
У. Обзоры, указатели
Ф. Информация

А РАЗРАБОТКИ УРОКОВ И ТЕМ; МАТЕРИАЛЫ ДЛЯ УРОКОВ И ФАКУЛЬТАТИВНЫХ ЗАНЯТИЙ

А.0.1

А.А. Зайцев. Повесть о настоящей черепашке. № 3

А.0.2

Г.В. Луканина. Графика в LOGO. Поурочная методическая разработка по информатике для класса УКНЦ. 7–8-е классы. № 5, 10

А.0.3

С.М. Окулов. Основы программирования. № 6, 7, 8, 10, 11, 13, 14, 16, 19, 23, 27

А.0.4

О.В. Титова. Полярные цветы. № 19

А.0.5

В.П. Гладков, А.П. Шестаков. Вопросы, задания и контрольные работы для начинающих программистов. № 20, 33, 34, 35, 37, 38, 40, 47, 48

А.0.6

Н.Г. Иванова, О.Л. Русакова. Информатика для школьников 5–6-х классов: ЛОГО-логика. № 28

А.0.7

Т.В. Бычкова. Учебно-игровой урок по теме “Алгоритмы и работа с ними”. № 36

А.0.8

Д.П. Береговой. Лабораторные работы по алгоритмизации. № 44, 45, 46

А.1. Как это делаю я

А.1.1

В.М. Нечаев. Сетевой вариант Microsoft Office. Работа в локальной сети Windows 95. № 8

А.1.2

А.Л. Королев. Как мы организовали школьный Интранет. № 42

А.2. Семинар

А.2.1

Г.Н. Гутман. Четыре занятия по ООП. № 5

А.2.2

А.А. Казанский, Л.В. Ларина. Перечисление булевых функций. № 15

А.2.3

Д.М. Златопольский. Элементы управления в программах на Visual Basic. № 16, 17, 18

А.2.4

А.А. Дуванов. JavaScript-конструирование. Материалы Роботландского университета.

Выпуск 1. № 21

Выпуск 2. № 25

Выпуск 3. № 29

Выпуск 4. № 33

А.2.5

Лев Наумов. Как увеличить скорость “Жизни”, или Эффективная организация данных для повышения скорости поиска клеток и разрешения отношений соседства при реализации клеточного автомата Джона Хортон Конвея “Жизнь”. № 33, 34

А.2.6

Е.А. Еремин. Компилятор? Это довольно просто! № 40, 43, 45, 46, 47

А.3. Урок на час

А.3.1

В.С. Голубев. Представление информации, системы счисления и основы логики. № 9

А.4. Методика

А.4.1

И.Н. Фалина. Работа в командах при изучении информационных технологий. № 6

А.4.2

Е.В. Андреева. Принципы проверки учебных и олимпиадных задач по информатике. № 34

А.4.3

Е.В. Андреева. Примерное содержание компьютерного задачника по информатике. № 35

А.4.4

Е.В. Андреева, И.С. Гуцин. Логические выражения. Практикум по информатике. № 36

А.4.5

Е.П. Паклина. Развитие учащихся на уроках информатики. № 46

А.5. Внеклассная работа по информатике; кроссворды, ребусы

А.5.1

Д.М. Златопольский. Внеклассная работа по информатике. Сборник заданий с решениями. № 4

Внеклассная работа по информатике. № 7, 11, 13, 14, 19, 20, 44

Кроссворд "Текстовый редактор Microsoft Word". № 10

Чайнворды. № 15

Венгерские головоломки. № 16

Собрать правильный шестиугольник. № 34

Кроссворд с фрагментами. № 37, 38

Учебное лото по информатике. № 43
± БУКВА = НОВОЕ СЛОВО. № 47

A.5.2

Т.П. Алексеева, И.В. Котегова. Интеллектуальная игра. № 36

A.5.3

А.В. Левина. Учебный кроссворд по информатике. № 47

A.6. Калейдоскоп

A.6.1

№ 33

Прогноз погоды к завтраку

Премия за борьбу с кошками

История компьютеров

Самая крупная махинация с использованием компьютера

№ 34

Компьютеры играют на бирже лучше людей

Грибы, но несъедобные

№ 37

Больше информации

Вредоносный компьютер

№ 38

Интернет в помощь

Отпечатки пальцев для учета рабочего времени

Виртуальный рентген

№ 39

Компьютер не пахнет?

Компьютер станет чувствовать чужое настроение

№ 40

150 Гб винчестер через три года

Шифр текста

"Компьютерное бешенство" в великой Британии

Компьютеризованная одежда

№ 43

Электроника на службе у незрячих

Кошелек с начинкой

№ 44

"В Космос будут ездить на лифте..."

Гениальная статистика

№ 45

Компьютерные и кинематографические миры

О поиске в Интернете

№ 46

Интернет в лифте

№ 47

Хопботы — кто они такие?

Б ЗАДАЧИ, ЗАДАЧИ...

Б.0.1

О.Л. Гусева, В.Ю. Гусев, Н.Н. Миронova. Одна задача — два подхода к решению. № 6

Б.0.2

О.В. Туркин. Использование электронных таблиц при исследовании физических моделей. № 17

Б.0.3

Д.М. Златопольский. Решение задач по теме "Целочисленная арифметика" с помощью электронных таблиц. № 43

Б.0.4

Д.М. Златопольский. Решение задач на обработку текста в электронных таблицах. № 44

Б.0.5

Д.М. Златопольский. Работа с данными типа "дата" в электронной таблице Microsoft Excel. № 47

В КОМПЬЮТЕРНЫЕ ВИРУСЫ И ЗАЩИТА ОТ НИХ

В.0.1

С.Л. Островский. Компьютерные вирусы. № 32

Г НАЧАЛЬНАЯ ШКОЛА

Г.0.1

Ю.А. Первин. Зимние вечера. Информатика для начинающих. Выпуски 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
Материалы Роботландского университета. № 1, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18

Д ИНФОРМАТИКА ДЛЯ МАЛЫШЕЙ

Д.0.1

Н.Н. Булгаков. Знакомство с компьютером в детском саду. № 18

Е ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Е.0.1

А.В. Орлов. Интернет-телефония. № 47

Е.1. Беседы

Е.1.1

А.А. Дуванов. "Назаметки" Сидорова. № 6, 7, 8, 9, 10, 11, 13, 15, 16, 17, 18, 19, 20

Е.2. Узелки на память

Е.2.1

С.Г. Никитенко. Интернет для учителя информатики. № 16, 17, 18

Е.2.2

Н.П. Заходняя. Эрмитаж (<http://www.hermitage.ru>). № 37

Ж ОБЩИЕ ВОПРОСЫ ПРЕПОДАВАНИЯ ИНФОРМАТИКИ

Ж.0.1

А.Г. Гейн. Обязательный минимум содержания образования по информатике: и в нем нам хочется дойти до самой сути. № 24, 30, 35, 36, 38, 39, 41, 42

Ж.1. Учебники, учебные пособия, словари

Ж.1.1

А.В. Орлов, А.И. Сенокосов. Словарь по информационным технологиям. № 2

Ж.1.2

English for Computer Science Teachers (Английский для учителя информатики). № 31, 34, 37, 41, 42, 43

З ИСТОРИЯ НАУКИ, ИСТОРИЯ ИНФОРМАТИКИ

З.0.1

Великий труженик. № 5

З.0.2

Д.М. Златопольский. Вычислительная машина Шиккарда. № 33

З.0.3

Д.М. Златопольский. Последователи Джона Непера. № 38

И ЮБИЛЕИ И ПАМЯТНЫЕ ДАТЫ

И.0.1

Ровесник Фортрана. № 6

И.0.2

Очень важное открытие. № 7

И.0.3

Подданный России. № 8

И.0.4

Мультимедийный процессор. № 9

И.0.5

Георг Ом и его закон. № 10

И.0.6
Двадцатилетие. № 11

И.0.7
Шарль Кулон. № 13

И.0.8
Рождение электроники. № 14

И.0.9
"Третий кит". № 15

И.0.10
Лидер отечественного программирования. № 16

И.0.11
Штурм, только мозговой. № 17

И.0.12
Изобретение века. № 18

И.0.13
Величины, единицы, системы. № 19

И.0.14
Фарадей. № 20

И.0.15
Юбилей. № 34

И.0.16
Врач, изобретатель, математик. № 35

И.0.17
Ровесница IBM PC. № 36

И.0.18
Мыслитель. № 37

И.0.19
От ARPAnet к Internet. № 38

И.0.20
Ученый, инженер, изобретатель. № 39

И.0.21
(Lots of (Idiotic (Silly (Parentheses))))).
№ 40

И.0.22
Открывший радиоволны. № 41

И.0.23
Система для передачи звука. № 42

И.0.24
Машины для письма. № 43

И.0.25
"Неограниченный алмаз". № 44

И.0.26
От второго к третьему. № 45

И.0.27
...и русские счета. № 47

И НА СТЕНД В КАБИНЕТЕ
ИНФОРМАТИКИ

К.0.1
Принцип томографии. № 5

К.0.2
"Минск-32". № 6

К.0.3
Декарт. № 7

К.0.4
В поисках эффективности. № 8

К.0.5
Леди Лавлейс. № 9

К.0.6
Holos + ...графия. № 10

К.0.7
"Врожденный механик". № 11

К.0.8
Машина М-3. № 13

К.0.9
Оптический, или семафорный. № 14

К.0.10
Джон Непер. № 15

К.0.11
Сначала — релейные. № 16

К.0.12
Пентиумы. № 17

К.0.13
Василий Петров. № 18

К.0.14
Предшественники. № 19

К.0.15
Пионеры. № 20

К.0.16
Евдокс. № 26

К.0.17
Всего лишь линейка. № 26

К.0.18
Христиан Гюйгенс. № 26

К.0.19
Механические калькуляторы. № 26

К.0.20
На пути к телеграфу. № 26

К.0.21
Американец. № 26

К.0.22
Создание электродинамики. № 26

К.0.23
Знаменитый англичанин. № 26

К.0.24
Комплексы и комплекты. № 26

К.0.25
Кирхгоф. № 26

К.0.26
Лорд Кельвин. № 26

К.0.27
Речной, а затем морской. № 26

К.0.28
Мастер эксперимента. № 26

К.0.29
Глава русских физиков. № 26

К.0.30
Первый трансатлантический. № 26

К.0.31
Вундеркинд. № 26

К.0.32
Конструктор. № 26

К.0.33
Просто электронный вычислитель.
№ 26

К.0.34
Atanasoff Berry Computer. № 26

К.0.35
Семейство "Марк". № 26

К.0.36
Первая в Европе. № 26

К.0.37
"Киев", "Днепр" и другие. № 26

К.0.38
Машина "Сетунь". № 26

К.0.39
Ставшие классическими. № 26

К.0.40
Язык Ада. № 26

К.0.41
Чипы. № 26

K.0.42

Утилиты, утилиты... № 26

K.0.43

Попробуйте упаковать. № 26

K.0.44

Первый инженер России. № 34

K.0.45

Болонья, Париж, ..., Москва, ... № 35

K.0.46

Бейсик — это просто? № 37

K.0.47

Почти первый. № 38

K.0.48

Мини-компьютеры: PDP и другие. № 40

K.0.49

Преемник Алгола. № 41

K.0.50

Не только транзисторы. № 42

K.0.51

Excel: путь к успеху. № 43

K.0.52

Компактный и легкорасширяемый. № 44

K.0.53

Борец за дисциплину в программировании. № 47

K.0.54

Уильям Бэрроуз. № 48

□ ПРЕДЛАГАЮ КОЛЛЕГАМ

A.0.1

И.Г. Гладких. Сделайте среду программирования более комфортной. № 5

A.0.2

Б.П. Сайков. Excel для любознательных. № 7, 8, 9

A.0.3

С.Л. Жаров. Опыт изучения машины Поста. № 10

A.0.4

Н.В. Бакшаева. Графика для web-странички. № 37

A.0.5

Е.А. Еремин. Имитатор поисковой машины как эффективное средство обучения поиску информации в Интернете. № 45

■ ОЛИМПИАДЫ, КОНФЕРЕНЦИИ, ТУРНИРЫ, ШКОЛЫ, КУРСЫ, СЕМИНАРЫ, ВЫСТАВКИ

M.0.1

№ 12. Олимпиады по информатике
С.М. Окулов, Д.С. Шулятников. Разбор задач международной олимпиады 2000 года.

А.С. Станкевич. Решение задач I Всероссийской командной олимпиады по программированию.

Итоги международной олимпиады по информатике 2000 года. Распределение медалей между странами

M.0.2

В.М. Кирюхин. Убедительная победа российских школьников на XII международной олимпиаде по информатике. № 14

M.0.3

А.С. Пономаренко. Пятая олимпиада по программированию в ЦАО г. Москвы. № 16

M.0.4

В.М. Кирюхин. XIII Всероссийская олимпиада школьников по информатике. № 18

M.0.5

Е.В. Андреева. Задачи XIII Всероссийской олимпиады по информатике. № 19

M.0.6

№ 22. Рыбинские городские школьные олимпиады по информатике

В.Н. Пинаев. Олимпиады по информатике. Методические рекомендации учителям при подготовке олимпиад по информатике. Анализ олимпиадных заданий

M.0.7

Л.А. Залогова, С.В. Русаков, И.Г. Семакин, Л.В. Шестакова. Пермская олимпиада по базовому курсу информатики. № 33, 34, 35

M.0.8

В.М. Кирюхин. XIII международная олимпиада школьников по информатике. № 37

M.0.9

Е.В. Андреева. Решение задач XIII международной олимпиады. № 37, 40, 42, 43, 44

M.0.10

Д.Ю. Усенков. ИТО-2001: традиции и новинки выставки. № 46

■ ОФИЦИАЛЬНЫЕ И НЕОФИЦИАЛЬНЫЕ ДОКУМЕНТЫ

H.0.1

Требования к уровню подготовки выпускников по информатике. Обязательный минимум содержания образования по информатике. (Проект). № 44

□ МНЕНИЯ

O.0.1

А.Ю. Уваров. Компьютеризация-2001. № 14

O.0.2

А.А. Кузнецов, А.Л. Семенов, А.Ю. Уваров. О проекте концепции образовательной области "Информатика и информационные технологии". № 17

O.0.3

"Оставьте информатику школе, а какой ей быть — отрегулирует сама жизнь". № 19

O.0.4

А.Н. Петров. Переворот в сознании: разрушительные идеи. № 20

O.0.5

А.Л. Семенов, А.Ю. Уваров. Переворот в сознании? № 20

O.0.6

А.В. Климачков. Учи, зарабатывай, учи. № 47

■ СТРАНИЦЫ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

P.0.1

И.Н. Фалина. Современные педагогические технологии и частные методики обучения информатике. № 37, 39, 41, 43, 45, 47

P.0.2

Е.В. Андреева. Олимпиады по информатике. Пути к вершине. № 38, 40, 42, 44, 46, 48

■ "КНИЖНЫЙ ШКАФ" УЧИТЕЛЯ ИНФОРМАТИКИ

P.0.1

Классические учебники: Computer Science. Алгоритмы. Построение и анализ (Т.Кормен, Ч.Лейзерсон, Р.Ривест). Алгоритмы. Построение и анализ: Пер. с англ. М.: МЦНМО, 2000). № 5

P.0.2

За границами предмета (В.И. Карлашук. Обучающие программы. М.: Солон-Р, 2001). № 7

P.0.3

Освой самостоятельно Visual Basic 5 (Н.Гуревич, О.Гуревич. Освой самостоятельно Visual Basic 5. Путеводитель по миру визуального программирования на языке Visual Basic 5. М.: ЗАО "Издательство "Бином", 1998). № 19

P.0.4

Использование "фотокамеры фантазии" (Д.Джордан, С.Мониц. Использование Adobe Photoshop 5. Основные навыки, простые решения. Издательский дом "Вильямс", 1999). № 20

P.0.5

Структурное проектирование и конструирование программ (Э.Йодан. Структурное проектирование и конструирование программ. М.: Мир, 1979). № 33

P.0.6

Как работают компьютеры? (Чарльз Петцольд. Код. М.: Издательско-торговый дом "Русская редакция", 2001). № 34

P.0.7

Алексей Андреевич Ляпунов. Составители: Н.А. Ляпунова, Я.И. Фет. Новосибирск, Изд-во ИВМ и МГ СО РАН, 2001. № 48

Н НАЧНИТЕ С ПРОСТОГО**T.0.1**

Е.Ефимов. Простая программа для решения квадратных уравнений на языке Visual Basic. № 46

О ОБЗОРЫ, УКАЗАТЕЛИ**У.0.1**

Тематический указатель материалов, помещенных в газете "Информатика" в 2001 году (№ 1—48). № 48

И ИНФОРМАЦИЯ**F.0.1**

Примерные правила для авторов газеты "Информатика". № 2, 10, 16, 22
Форма для выплаты гонорара за публикации. № 2, 7, 8, 16, 22

F.0.2

Проект для учителей информатики "Программист-2001". Дистанционная обучающая олимпиада и семинар по программированию-2001. № 3

F.0.3

Приглашение к участию в конкурсе по созданию учебной литературы нового поколения. № 6, 7

F.0.4

Литература Объединения педагогических изданий "Первое сентября". № 7, 11, 22, 23, 28, 30, 37, 42, 45

F.0.5

Телетестинг. № 9

F.0.6

Роботландский сетевой университет. Сетевые курсы обучения учителей и школьников. Предварительный набор на 2001/2002 учебный год. № 17

Роботландский университет. № 34

F.0.7

А.А. Дуванов. Роботландский сетевой университет глазами учителей и школьников. № 34

F.0.8

Вторая конференция "Информатика". № 18

F.0.9

XI международная юбилейная конференция "Информационные технологии в образовании" ("ИТО-2000"). № 27, 32, 33, 40, 46

F.0.10

Что "Информатика" предложит подписчикам в 2001/2002 учебном году. № 38, 40, 42, 44, 45

F.0.11

Издательский дом "Первое сентября". № 41, 43

F.0.12

Пятые Соловейчиковские чтения. № 32, 34

F.0.13

Урок толерантности. № 36, 39

Ч ЧИТАЙТЕ В ЯНВАРСКИХ НОМЕРАХ**C.0.1**

Анонс нового Роботландского курса информатики для младших школьников "Азы информатики". № 39

C.0.2

А.А. Дуванов, Ю.А. Первин. Тропинка конструктора. Новые встречи с дедушкой Фёрстовым. № 41

Предметно-именной указатель (имена, термины, названия и ссылки на статьи)

А

Абак — И.0.27
Автомат клеточный — А.2.5
Авторское право — Е.1.1
Ада (язык программирования) — К.0.40, К.0.49
Азы информатики — С.0.1
АИСТ — И.0.10
Аксиома Архимеда — см. Архимеда аксиома
Алгол — К.0.39, К.0.49
Алгоритм — А.0.7, А.0.8, Р.0.1
Алгоритм поиска — П.0.2
Алгоритм разветвляющийся — А.0.8
Алгоритм циклический — А.0.8
Аллен Пол — К.0.46
Алферов Жорес Иванович — К.0.41
Альтаир-8800 — И.0.9, К.0.14, К.0.46
АЛЬФА — И.0.10
АЛЬФА-6 — И.0.10

Ампер Андре Мари — К.0.22
Аналитическая машина — К.0.5, К.0.11, К.0.23, К.0.35
Английский язык — Ж.1.2
Английский язык для учителя информатики — Ж.1.2
Анимация компьютерная — Е.1.1
Аннигиляция — К.0.1
Антиалиасинг — Е.1.1
АПТ — И.0.1
Арифмометр — К.0.19, К.0.54
Арифмометр "Феликс" — К.0.19
Арифмометр Чебышева — см. Чебышева арифмометр
Архиватор — см. Программа-упаковщик
Архимеда аксиома — К.0.16
Атанасов Джон Винсент — К.0.34, К.0.47

Б

База данных — И.0.9
Бардин Джон — И.0.12
Барнэби Джон — К.0.14
Бейсик — Б.0.1, К.0.46, К.0.53, Л.0.1
Белл Александер Грейам — И.0.23
Бернулли Даниил — К.0.7
Бернулли Иоганн — И.0.18
Бернулли Якоб — И.0.18
Берри Клиффорд — К.0.34
Бессонов Николай Иванович — К.0.11
БЕТА — И.0.10
Браттейн Уолтер — И.0.12
Браузер — Е.1.1
Бриклин Дэниел — К.0.14
Брук Исаак — К.0.8
Брусенцов Н.П. — К.0.38
Булева функция — см. Функция булева
Бурсель Шарль — И.0.23
Бэббидж Чарльз — К.0.5, К.0.23, К.0.35

Бэкус Джон — К.0.39
 Бэрроуз Уильям — К.0.54
 БЭСМ — К.0.32
 БЭСМ-6 — К.0.32

В

Ввод-вывод — А.0.8
 Веб-графика — Е.1.1, Л.0.4
 Веб-дизайн — Е.1.1, Л.0.4
 Веб-сервер — А.1.1
 Веб-страница — Е.1.1, Л.0.4
 Вебер Вильгельм Эдуард — И.0.13
 Винклер Иоганн Генрих — К.0.20
 Вирт Никлаус — А.0.5, К.0.40, К.0.49
 Вирус — см. Компьютерный вирус
 Внеклассная работа — А.5.1, А.5.2
 Вопрос — А.0.5
 Второе поколение компьютеров — см. Компьютер второго поколения
 Выставка "Школа-2001" — Ф.0.8
 Вычислительный центр Академии наук СССР (ВЦ АН СССР) — И.0.10
 Вычислительный центр Сибирского отделения Академии наук СССР (ВЦ СО АН СССР) — И.0.10

Г

Габор Деннис — К.0.6
 Газета "Информатика" — Ф.0.10
 Гальванопластика — И.0.3
 Гаусс Карл Фридрих — И.0.13
 Гейтс Билл — К.0.46
 Герике Отто — см. Фон Герике Отто
 Герц Генрих — И.0.2, И.0.22, К.0.22, К.0.29
 Гильберт Уильям — К.0.15
 Гитторф Иоганн — И.0.2
 Глиден Карлос — И.0.24
 Глушков Виктор Михайлович — К.0.37
 "Голиаф" (судно) — К.0.27
 Головоломка — А.5.1
 Голограмма — К.0.6
 Голография — К.0.6
 График функции — А.0.4
 Графика компьютерная — А.0.4, Е.1.1, Л.0.4
 Графика черепашня — см. Черепашня графика
 Грей Стефан — К.0.15
 Грей Элиша — И.0.23
 "Грейт Истерн" (судно) — К.0.30
 Гук Роберт — К.0.9, К.0.18
 Гюйгенс Христиан — И.0.18, К.0.18

Д

Дедушка Фёрстов — Г.0.1, С.0.2
 Декарт — К.0.3
 Денисюк Юрий Николаевич — К.0.6
 Диагностика компьютерная — К.0.1
 Дijkstra Эдсгер — К.0.53
 "Днепр" — К.0.37
 Документ — Н.0.1
 Дошкольник — Д.0.1

Е

"Е97" — см. Учебный процессор "Е97"
 Евдокс — К.0.15
 Ершов Андрей Петрович — И.0.10
 ЕС ЭВМ — И.0.26, К.0.4

Ж

"Жизнь" — см. Игра "Жизнь"

З

Задача — А.0.5, А.4.3, Б.0.1 — Б.0.5, М.0.1, М.0.3, М.0.5 — М.0.7, М.0.9, П.0.2
 Задача комбинаторная — П.0.2
 Задача олимпиадная — М.0.1, М.0.3, М.0.5 — М.0.7, М.0.9, П.0.2
 Задачи проверки решения — А.4.2
 Задачник — А.4.3
 Задачник компьютерный — А.4.3
 Закон Кулона — см. Кулона закон
 Закон Ома — см. Ома закон
 Законы Ньютона — см. Ньютона законы
 "Зимние вечера" — Г.0.1, С.0.2

И

Игра — А.2.5, А.5.1, А.5.2
 Игра "Жизнь" — А.2.5
 Игра интеллектуальная — А.5.1, А.5.2
 "Игра 15" — А.2.1, Г.0.1
 Издательский дом "Первое сентября" — Ф.0.11
 Изобретение — К.0.7
 Инкапсуляция — А.2.1
 Институт точной механики и вычислительной техники (ИТМ и ВТ) — И.0.10
 Интегральная схема — И.0.26, К.0.4, К.0.41
 Интернет — Е.0.1, Е.1.1, Е.2.1, Е.2.2, И.0.19, И.0.23, Л.0.5
 "Интернет для учителя" — см. Справочник "Интернет для учителя"
 Интранет — А.1.2
 "Информатика" — см. газета "Информатика"
 Информатика для малышей — Д.0.1
 Информатики преподавание — А.4.1, А.4.2, О.0.1, О.0.2 — О.0.5, П.0.1, П.0.2
 Информационные технологии — А.4.1, Ж.1.1, Л.0.4, Р.0.6
 "Информационные технологии в образовании" — М.0.10, Ф.0.9
 ИПЛ — И.0.21
 Искусственный интеллект — И.0.21
 Искусство — Е.2.2
 Исполнитель — А.0.6, Г.0.1
 ИТМ и ВТ — см. Институт точной механики и вычислительной техники (ИТМ и ВТ) — И.0.10, К.0.32
 "ИТО-2001" — М.0.10, Ф.0.9
 Ихбиа Жан — К.0.40

К

Кабель телеграфный — К.0.27, К.0.30
 Кавендиш Генри — И.0.14, К.0.22
 "Калейдоскоп" — А.6.1
 Калькулятор Лейбница — см. Лейбница калькулятор
 Кардано Джероламо — И.0.16
 Кардано формула — И.0.16
 Карцев Михаил — К.0.8
 Качество обучения — П.0.1
 Кельвин (Уильям Томсон) — К.0.26
 Кемени Джон — К.0.46
 Кеплер Иоганн — З.0.2
 Керниган Брайан — И.0.15
 Кетков Юлий Лазаревич — К.0.46
 "Киев" — К.0.37
 Килби Джек — К.0.41
 Кирхгоф Густав Роберт — К.0.25
 Кирхгофа правила — К.0.25
 Класс компьютерный — см. Компьютерный класс
 Класс УКНЦ — см. УКНЦ класс

Клеточный автомат — см. Автомат клеточный
 Книга — Р.0.1—Р.0.7
 Книга гипертекстовая — С.0.1
 Кобол — К.0.39
 Коммерция — О.0.6
 Компилятор — А.2.6
 Компьютер — З.0.1, К.0.31, К.0.32, К.0.47, К.0.48
 Компьютер аналоговый — З.0.1
 Компьютер второго поколения — И.0.26, К.0.4, К.0.50
 Компьютер первого поколения — К.0.50
 Компьютер персональный — К.0.48
 Компьютер релейный — К.0.11
 Компьютер третьего поколения — И.0.26, К.0.4
 Компьютеризация школ — О.0.1
 Компьютерная графика — см. Графика компьютерная
 "Компьютерное конструирование" — С.0.2
 Компьютерный вирус — В.0.1
 Компьютерный класс — А.0.2, О.0.6
 Конкурс — Ф.0.3, Ф.0.13
 Контрольная работа — А.0.5
 Конференция — Ф.0.8, Ф.0.9
 Конференция "Информатики" — Ф.0.8
 Концепция образовательной области "Информатика и информационные технологии" (проект) — О.0.2
 Коперник Николай — К.0.18
 Кремер Герберт — К.0.41
 Кроссворд — А.5.1, А.5.3
 Крылов Алексей Николаевич — З.0.1
 Кулибин Иван Петрович — К.0.7, К.0.9
 Кулон Шарль Огюстен — И.0.7
 Кулона закон — И.0.7
 Курц Томас — К.0.46

Л

Лабораторная работа — А.0.8
 Лавлейс Ада Августа — К.0.5, К.0.40, К.0.49
 Лампа вакуумная — И.0.8, И.0.12
 Лашли Хэл — И.0.9
 Лебедев Петр Николаевич — К.0.28
 Лебедев Сергей Алексеевич — К.0.32, К.0.36
 Лейбниц Готфрид Вильгельм — И.0.18, К.0.18, К.0.19
 Лейбница калькулятор — И.0.18, К.0.18, К.0.19
 Лейт Эммет — К.0.6
 Ленард Филипп — И.0.2
 Ленц Эмилий Христианович — И.0.3
 Лисп — И.0.21
 Лисп-компьютер — И.0.21
 Логарифм — И.0.27, К.0.10
 Логарифмическая линейка — К.0.17
 Логика — А.0.6, А.3.1
 Логика аристотелева — см. Логика
 Логика математическая — А.3.1
 Логика формальная — см. Логика
 Логические выражения — А.4.4
 Лого — А.0.1, А.0.2, А.0.6
 Лого-система — А.0.1
 Лодыгин Александр Николаевич — К.0.13, К.0.22
 Ломоносов Михаил Васильевич — К.0.21, К.0.45
 Лоренц Хендрик Антон — И.0.22
 Лосев Олег Владимирович — И.0.12
 Лото учебное — А.5.1
 Лямбда-исчисление — И.0.21
 Ляпунов Алексей Андреевич — И.0.10, Р.0.7

М

М-3 — К.0.8
 М-20 — К.0.32, К.0.46
 Магнит — К.0.15
 Макинтош — К.0.51
 Маккарти Джон — И.0.21
 Макрос — Л.0.1
 Максвелл Джеймс Клерк — И.0.14, И.0.22, К.0.22
 “Марк” (семейство компьютеров) — К.0.35
 Маркони Гульельмо — К.0.22
 Марчук Гурий Иванович — И.0.10
 Маучли Джон — И.0.12, К.0.47
 Машина Поста — см. Поста машина
 Машина Тьюринга — см. Тьюринга машина
 Машина Шиккарда — см. Шиккарда машина
 Машинная графика — см. Графика компьютерная
 МГУ — см. Московский государственный университет им. М.В. Ломоносова
 Международная система единиц (СИ) — И.0.13
 Метод мозгового штурма — см. Мозгового штурма метод
 Методика преподавания — А.4.1, А.4.4, А.4.5, П.0.1
 Методика преподавания в начальной школе — П.0.1
 Микропроцессор — И.0.25, К.0.12
 Микропрограммирование — И.0.26, К.0.4
 Мини-компьютер — К.0.48
 Министерство образования — О.0.2
 “Минск” — К.0.2
 “Минск-22” — К.0.2
 “Минск-32” — К.0.2
 МИР (“машина для инженерных расчетов”) — К.0.37
 Мнение — О.0.1 — О.0.6
 Модель физическая — Б.0.1
 Мозгового штурма метод — И.0.11
 Московский государственный университет им. М.В. Ломоносова — К.0.38, К.0.45
 Мультипликация компьютерная — см. Анимация компьютерная
 Мур Чарлз — К.0.52
 Мышление — А.0.6
 МЭСМ — К.0.32, К.0.36

Н

Навигация по сайту — Е.1.1
 Наследование — А.2.1
 Натворд — А.5.1
 Начальная школа — Г.0.1
 Нейман Джон — см. Фон Нейман Джон
 Непер Джон — З.0.3, К.0.10, К.0.17
 Непера палочки — З.0.3, И.0.27, К.0.10
 Непера последователи — З.0.3
 Нойс Роберт — К.0.41
 Нортон Питер — К.0.42
 Ньютон Исаак — И.0.18
 Ньютона законы — З.0.1

О

Обучение сетевое — С.0.1, Ф.0.6
 Объект (в объектно-ориентированном программировании) — А.2.1
 “Обязательный минимум содержания образования по информатике” — А.4.3, Ж.0.1, Н.0.1
 Однер Вильгодт (петербургский инженер) — К.0.19
 Однера колесо — К.0.19

Олимпиада — М.0.1 — М.0.9, П.0.2, Ф.0.2
 Олимпиады правила проведения — П.0.2
 Ом Георг — И.0.5, К.0.13
 Ома закон — И.0.5
 Омонимы (слова, имеющие одинаковое звучание, но разные значения) — А.5.1
 ООП — см. Программирование объектно-ориентированное
 Оптимизация — Б.0.1

П

Память буферная — И.0.24
 “Паскалево колесо” — см. Паскаля суммирующая машина
 Паскаль (язык программирования) — А.0.3, А.0.5, А.2.1, А.2.6, Б.0.1, К.0.40, К.0.49, К.0.53, Л.0.1
 Паскаль Блез — К.0.19
 Паскаля суммирующая машина — И.0.18, К.0.19
 Пейперт Сеймур — О.0.1
 Первое поколение компьютеров — см. Компьютер первого поколения
 Петр I — К.0.45
 Петров Василий Владимирович — К.0.13
 “Пилотные школы” — А.0.1
 Пишущая машинка — И.0.24
 Планкалькуль — К.0.11
 Плюккер Улиус — И.0.2
 Повышение квалификации — П.0.1, П.0.2
 Поиск информации — А.0.3, Л.0.5
 Поиск в Интернете — Л.0.5
 Полиморфизм — А.2.1
 Полупроводник — И.0.12
 Попов Александр Степанович — И.0.20, И.0.22, К.0.22
 Поста машина — Л.0.3
 Правила для авторов — Ф.0.1
 Правила Кирхгофа — см. Кирхгофа правила
 Правила проведения олимпиады — см. Олимпиады правила проведения
 Практикум — А.4.4
 Преподавание информатики — см. Информатики преподавание
 Принтер — И.0.24
 Проверка решения задачи — см. Задачи проверка решения
 Программа обучающая — П.0.1, Р.0.2
 Программа-упаковщик — К.0.43
 Программирование — А.0.3, А.0.5, А.2.1, К.0.53, Р.0.3, Т.0.1
 Программирование объектно-ориентированное — А.2.1
 Программирование структурное — К.0.53, Р.0.5
 Проект — Ф.0.2, Ф.0.13
 Пролог — И.0.21
 “Проминь” — К.0.37
 Процедура — А.0.2
 Процессор — см. Микропроцессор

Р

Развитие учащихся — А.4.5
 Разделение времени — И.0.26, К.0.4
 Разностная машина — К.0.5, К.0.23
 Разер Элизабет — К.0.52
 RBM-1 (релейный компьютер) — К.0.11
 Ребус — А.5.1
 Редактор текстовый — см. Текстовый редактор
 Режим реального времени — И.0.26, К.0.4
 Рейс Иоганн Филипп — И.0.23
 Рекурсия — А.0.3
 Релком — И.0.19
 Рисунок — А.0.4, Г.0.1

Ритчи Деннис — И.0.15
 Ричардсон Оуэн Уилланс — И.0.8
 Роботландский курс информатики — С.0.1
 Роботландский сетевой университет — А.2.4, С.0.1, С.0.2, Ф.0.6
 Роботландского сетевого университета материалы — А.2.4, Г.0.1, С.0.2
 Рональдс Фрэнсис — К.0.20
 Росс Дуглас — И.0.1
 Российский научный центр “Курчатовский институт” — И.0.19
 Рубинштейн Сеймур — К.0.14
 Рыбинск — М.0.6
 Рэтлифф Уэйн — И.0.9

С

Сайт — Г.0.1, Е.1.1, Е.2.2
 Сальва Франсиско — К.0.20
 Санкт-Петербург — Е.2.2
 Связь — Е.0.1
 Семинар — А.2.1, А.2.2, А.2.3, А.2.4, А.2.5, А.2.6
 Семинар для учителей — Ф.0.2
 Сетевое обучение — см. Обучение сетевое
 “Сетунь” — К.0.38
 Сеть компьютерная — А.1.1, А.1.2
 Сеть локальная — А.1.1, А.1.2, О.0.6
 Сжатие информации — К.0.43
 СИ — см. Международная система единиц (СИ)
 Си (язык программирования) — И.0.15
 Сидоров Иван — “веб-дизайнер” — Е.1.1
 Синонимы — А.5.1
 Система счисления — А.3.1
 Словарь — Ж.1.1
 СМ-3, СМ-4 — К.0.48
 Соловейчиковские чтения — Ф.0.12
 Соробан (счетный прибор) — И.0.27
 Спортовка — А.0.3, А.2.5, П.0.2
 Справочник “Интернет для учителя” — Е.2.1
 Стек — А.5.1, Г.0.1
 Столетов Александр Григорьевич — И.0.14, К.0.25, К.0.29
 Структурное программирование — см. Программирование структурное
 Суан-пан (счетный прибор) — И.0.27
 СУБД — И.0.9
 СУНЦ МГУ — школа-интернат им. А.Н. Колмогорова — А.4.4
 Счетно-перфорационная техника — К.0.24
 Счеты — И.0.27
 США — К.0.44

Т

Табулятор — К.0.24
 Тег — Е.1.1
 Тег IMG — Е.1.1
 Тейт Джордж — И.0.9
 Текстовый редактор — Г.0.1
 Телеграф — К.0.9, К.0.20, К.0.27, К.0.30
 Телеграф гармонический — И.0.23
 Телеграф оптический (семафорный) — К.0.9
 Телестинг — Ф.0.5
 Телефон — И.0.23
 Телефония компьютерная — Е.0.1
 Термоэлектронная эмиссия — см. Эмиссия термоэлектронная
 Тесла Никола — И.0.20
 Тестирование — А.0.7, Л.0.2, Ф.0.5
 Тестирование командное — А.0.7
 Техника программирования — П.0.2
 Технология педагогическая — П.0.1
 Технология MMX — И.0.4

"Титаник" — 3.0.1, K.0.52

Томограф — K.0.1

Томография — K.0.1

Томсон Джозеф Джон — И.0.2

Томсон Уильям — см. Кельвин (Томсон Уильям)

Торвалдс Линус — М.0.8

Транзистор — И.0.8, И.0.12, K.0.50

Трансляция программы — A.2.6

Требования к уровню подготовки выпускников по информатике (проект) — Н.0.1

Третье поколение компьютеров — см. Компьютер третьего поколения

Триод полупроводниковый — см. Транзистор

Турбо Паскаль — A.0.3, A.0.5, K.0.49, Л.0.1

Тьюринга машина — Л.0.3

У

Указатель тематический — У.0.1

УКНЦ класс — A.0.2

Университет — K.0.45

Упатниекс Юрис — K.0.6

Уравнение квадратное — Т.0.1

Урок на час — A.3.1

Утилита — K.0.42

Учебная литература — Ф.0.13

Учебник — A.2.4, Р.0.1

Учебник гипертекстовый — A.2.4

Учебный процессор "Е97" — A.2.6

Ф

Фарадей Майкл — И.0.14, K.0.22

"Феликс" — см. Арифмометр "Феликс"

Фёрстов — см. Дедушка Фёрстов

Физика — Б.0.2

Физическая модель — см. Модель физическая

Флитвуд Кромвелл — И.0.2

Фон Герице Отто — K.0.15

Фон Нейман Джон — K.0.31

Форма для выплаты гонорара за публикации — Ф.0.1

Формат GIF — E.1.1

Формат GPG — E.1.1

Формат PNG — E.1.1

Формула Кардано — см. Кардано формула

Форт — K.0.52

Фортран — K.0.39

Фотометрия — K.0.29

Фотоэлемент — K.0.29

Фотоэффект — И.0.22, K.0.29

Франклин Бенджамин — K.0.20, K.0.21

Фрэнкстон Роберт — K.0.14

Функция (математическое понятие) — A.2.2

Функция булева — A.2.2

Х

"Ханойские башни" — Г.0.1

Хевисайд Оливер — И.0.22

Хеширование — A.2.5

Холлерит Герман — K.0.24

Хофф Маршиан Эдвард — K.0.41

Ц

Цветок — A.0.4

Цузе Конрад — K.0.11

Ч

Чайнворд — A.5.1

Чебышев Пафнутий Львович — K.0.19

Чебышева арифмометр — K.0.19

Черепашка — A.0.1, A.0.2, A.0.6

Черепашья графика — A.0.1, A.0.2

Черч Алонзо — И.0.21

Чип — см. Интегральная схема

Ш

Шапп Клод — K.0.9

Шиккард Вильгельм — 3.0.2, K.0.19

Шиккарда машина — 3.0.2, K.0.19

Шиллинг Павел Львович — K.0.13, K.0.20, K.0.22

"Школа-2001" — см. Выставка "Школа-2001"

Школа начальная — см. Начальная школа

Шокли Уильям — И.0.12

Шоулс Кристофер — И.0.24

Шрейер Марк — K.0.14

Штибитц Джордж — K.0.11

Шухов Владимир Григорьевич — K.0.44

Э

ЭВ-80 (электронный вычислитель) — K.0.33

Эдисон Томас — И.0.8, И.0.20, K.0.22

Эйкен Говард — K.0.11, K.0.35

Эйлер Леонард — И.0.13, K.0.7

Экерт Преспер — K.0.47

Экскурсия по сайту — E.2.2

Электрон — И.0.2

Электронная почта — Г.0.1

Электронные таблицы — Б.0.1 — Б.0.5

Эмиссия термоэлектронная — И.0.8

ЭНИАК — K.0.34, K.0.47

Эрмитаж — E.2.2

Эрстед Ханс Кристиан — K.0.22

Я

Язык программирования — И.0.1, K.0.39

Язык программирования специализированный — И.0.1

Якоби Борис Семенович — И.0.3, K.0.22

Якоби Карл Густав — И.0.3

Якобиан — И.0.3

* * *

А

ABC (Atanasoff Berry Computer) — K.0.34

Adobe Photoshop — P.0.4

AMD — K.0.12

ANSI — И.0.15

APT — см. АПТ

ARPA — И.0.19

ARPAnet — И.0.19

AT&T — И.0.12, И.0.15

В

BASIC — см. Бейсик

Bell Telephone Laboratories — И.0.12, И.0.15

Burroughs — K.0.54

С

С — см. Си

Celeron — И.0.25, K.0.12

Computer Science — P.0.1

CP/M — И.0.17

Д

dBase II — И.0.9

DEC — K.0.48

DOS — И.0.17, K.0.42

Е

EDVAC — K.0.47

ENIAC — см. ЭНИАК

Excel — Б.0.1 — Б.0.5, K.0.51, Л.0.2

Ф

Forth — см. Форт

Н

HTML — A.2.4, E.1.1

І

IBM — И.0.6, И.0.26, K.0.4, K.0.39

IBM 360 — И.0.26, K.0.4

IBM 370 — И.0.26, K.0.4

IBM PC — И.0.6, И.0.17, K.0.51

IFIP — И.0.10, K.0.49

Inferno — И.0.15

Intel — И.0.4, И.0.25, K.0.12, K.0.41

Intel 4004 — K.0.41

Internet — см. Интернет

IPL — см. ИПЛ

Ј

Java — A.2.4

JavaScript — A.2.4

Л

Limbo — И.0.15

Linux — М.0.8

Lisp — см. Лисп

Logo — см. Лого

Lotus 1-2-3 — K.0.51

М

Macintosh — см. Макинтош

Microsoft — K.0.46

Microsoft Excel — см. Excel

Microsoft Office — A.1.1

Microsoft Word — см. Word

MITS — K.0.46

MMX — см. Технология MMX

MS-DOS — см. DOS

MultiPlan — K.0.51

Р

PDP-5, PDP-8, PDP-11 — K.0.48

Pentium — И.0.25, K.0.12

Pentium II — И.0.25, K.0.12

Pentium III — И.0.25, K.0.12

Pentium 4 — И.0.25, K.0.12

Pentium MMX — И.0.4, И.0.25, K.0.12

Pentium Pro — И.0.25, K.0.12

Plankalkul — см. Планкалькуль

Prolog — см. Пролог

Р

Relcom — см. Релком

С

System 360 — И.0.26, K.0.4

Т

Texas Instruments — K.0.41

Turbo Pascal — см. Турбо Паскаль

U

UNIVAC — И.0.24

UNIX — И.0.15

V

VisiCalc — И.0.9, K.0.14, K.0.51

Visual Basic — A.2.3, P.0.3, T.0.1

VTG (программно-аппаратный комплекс) — E.0.1

W

Web-графика — см. Веб-графика

Web-дизайн — см. Веб-дизайн

Web-сервер — см. Веб-сервер

Web-страница — см. Веб-страница

Windows — K.0.51, И.0.17

Word — A.5.1

WordStar — И.0.9, K.0.14

X

XARA Ltd. — Л.0.4

Z

Z1, Z2, Z3, Z4, Z5 (компьютеры) — K.0.11

“Информатика в начальной школе”

К вопросам преподавания информатики в начальной школе наша газета обращается часто. На протяжении нескольких лет мы публиковали материалы, подготовленные различными авторскими коллективами, которые работают с учениками начальной школы. У нас даже были публикации, посвященные занятиям информатикой с ребятами в детском саду.

Говоря о материалах этой рубрики, нельзя не отметить две крупные публикации последнего года. Во-первых, в серии “Жаркое лето-2000” были опубликованы материалы пермского автора О.Л. Русаковой “Информатика. Уроки развития”. А в зимних и весенних номерах 2001 года был опубликован большой курс “Информатика для начинающих. Зимние вечера”. Его автор — Ю.А. Первин хорошо знаком нашим читателям. Думается, что нет ни одного преподавателя информатики в стране, который бы не слышал о “Роботландии” и ее создателях. “Зимние вечера” подготовлены на основе материалов Роботландского университета, с которым наша газета давно и плодотворно сотрудничает.

“Предлагаю коллегам”

У каждого из нас в процессе работы появляются собственные изящные решения типичных проблем, накапливается методический, организационный, а иногда и программистский опыт. Изобретать велосипеды, конечно, интересно (а иногда и просто необходимо), но в условиях школьной тесноты не слишком продуктивно. В рубрике “Предлагаю коллегам” публикуются рецепты решения различных вопросов, которыми делятся наши коллеги.

“Как это делаю я”

Популярная рубрика “Как это делаю я” появилась в нашей газете много лет назад. Идею организации такой рубрики предложил методист Московского института повышения квалификации работников образования В.М. Нечаев. Он же и стал основным автором материалов рубрики. Эти материалы очень популярны, ибо в них описываются типичные проблемы, с которыми каждому из нас приходится сталкиваться в повседневной работе. Авторы статей рубрики честно проходят весь путь — от формулировки проблемы до ее решения и описывают свой опыт на страницах газеты.

“Теоретические основы информатики”

Информатика продолжает оставаться молодым школьным предметом. Молодым не только по духу, что, наверное, будет всегда, но и организационно. Не все преподаватели информатики имеют фундаментальное образование в своей области. А это становится все более необходимо. Материалы рубрики “Теоретические основы информатики” призваны оказать помощь тем, кто хочет глубже познакомиться с научными основами нашего предмета. Первая же публикация в этой рубрике стала необыкновенно популярной. Речь идет о тематических выпусках “Земля информатика” (А.Г. Гейн), которые вышли летом 1996 года. Одна из последних публикаций — материал, в котором раскрываются положения “Обязательного минимума содержания образования по информатике”. Этот цикл статей называется “Обязательный минимум содержания образования по информатике: и в нем нам хочется дойти до самой сути”.

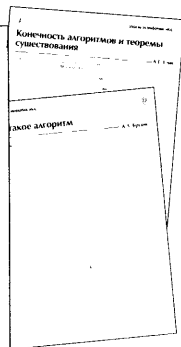
“Проекты”

Рубрика “Проекты” — особая. У редакции есть любопытный критерий, руководствуясь которым мы отбираем материалы для этой рубрики. Наверное, многие знают замечательную книгу Ч.Уэзерелла “Этюды для программистов”. Критерий отбора материалов в рубрику “Проекты” такой: а подходит ли материал для книги Уэзерелла? Разумеется, речь идет не о содержании книги, она все же рассчитана на подготовку программистов, а о духе и стиле соответствующих материалов. В рубрике “Проекты” опубликовано много интересных статей. Все описанные проекты были реализованы авторами (это является обязательным условием публикации), поэтому мы не просто предлагаем читателю идеи, “что в принципе можно сделать”, а рассказываем об опыте реализации конкретных проектов.

“Не только информатика”



В этой рубрике пока был опубликован всего... один материал. Зато очень хороший. В 2000 году мы отдали целый майский номер (№ 20/2000) нашему коллеге — учителю информатики из Иркутска Григорию Рейнгольду. В этом номере была опубликована его повесть “Школа выживания”.

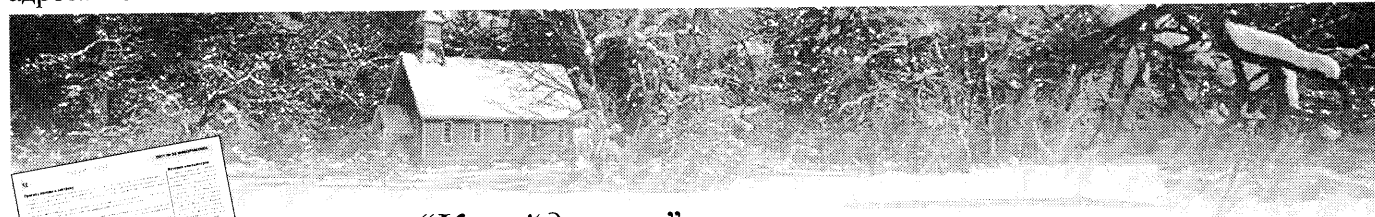
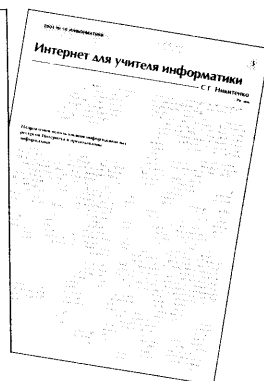


“На стенд в кабинете информатики”

Идея этой рубрики была подсказана одним из наших читателей, который посетовал на то, что при оформлении кабинета приходится перепечатывать, вырезать и переклеивать материалы газеты. И время на это лишнее уходит, и газету портить жалко. Поскольку все мы работаем в школе и сами сталкиваемся с необходимостью оформлять кабинеты, сама собой пришла мысль облегчить жизнь себе и читателям и публиковать материалы, уже готовые к тому, чтобы сразу использовать их на стендах. Когда рубрика эта только обсуждалась, в шутку даже был придуман простой принцип: “сегодня в газете — сегодня на стенде”. Рубрика “На стенд в кабинете информатики” оказалась такой удачной, что по итогам ежегодного анкетирования читателей именно она уверенно заняла первое место. В частности, по этой причине в серии “Жаркое лето-2001” вышел тематический выпуск, целиком составленный из материалов “на стенд”. Первые рейтинги летних номеров показывают, что и в этот раз материалы тематического номера пользуются большой популярностью. Поскольку практически все материалы рубрики “На стенд в кабинете информатики” готовятся в редакции, они, как правило, выходят без указания автора. На самом деле автор у них, конечно, есть. Причем все материалы “на стенд” готовит один человек — наш замечательный редактор Л.Н. Картвелишвили.

“Узелки на память”

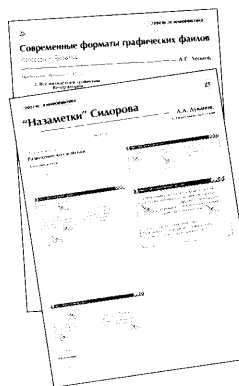
В рубрике “Узелки на память” мы знакомим наших читателей с интересными Интернет-ресурсами. Разумеется, те, кто постоянно работают в Сети, как правило, и “сами с усами”. Им нет необходимости читать пространные описания сайтов — они сами на них сходят, сами все посмотрят и оценят. Но и те, кто много работает в Сети, знают — часто мы не знаем о существовании того или иного ресурса не потому, что не можем его найти, а потому, что нам не приходит в голову его поискать. Приведем только один пример: оказывается, далеко не все знают об отличном орфографическом словаре на www.gramota.ru и замечательном онлайн-переводчике на “Яндексе” lingvo.yandex.ru. И такого рода ресурсов множество. Таким образом, в материалах этой рубрики читатели не найдут подробных описаний Интернет-ресурсов, мы не видим в этом необходимости. Наша задача — обратить внимание на любопытные и полезные адреса в Сети.



“Калейдоскоп”

“Калейдоскоп” — наша новая рубрика, она появилась в “Информатике” только осенью 2001 г. Появилась она в значительной степени в результате злоупотребления сотрудниками редакции своим служебным положением ☺.

Допустим, вы узнали что-то такое интересное, что вас “просто распирает” — хочется поделиться с кем-нибудь. Вы можете позвонить коллегам, написать, в конце концов разместить эту информацию на своем сайте в Интернете. А мы можем опубликовать ее в газете. Разумеется, материалы “Калейдоскопа” проходят придирчивую проверку на достоверность, мы всегда указываем источник информации, с тем чтобы наши читатели могли уверенно делиться интересными фактами с коллегами и учениками.



“Беседы”

Рубрика “Беседы” началась с публикации статей А.Г. Леонова “Современные форматы графических файлов. Вечера с Вовой”. Название этой рубрики продиктовано жанром публикуемых в ней материалов. Все они построены на основе бесед автора с воображаемым (как правило, имеющим реальный прототип) собеседником. Весной 2001 года в этой рубрике был опубликован получивший широкую известность цикл статей нашего постоянного автора А.А. Дуванова “Назаметки” Сидорова”.

ИНФОРМАТИКА НОВОГО ВЕКА (январь 2001 г.)

Ю.А. Первин.
Зимние
вечера.
Информатика
для начинающих



В этом номере началась публикация новой книги Ю.А. Первина, автора, хорошо знакомого всем, кто связан со школьной информатикой. Главы этой книги (вечера) публиковались в зимних и весенних номерах 2001 г.

А.В. Орлов,
А.И. Сенокосов.
Словарь по
информационным
технологиям



В словаре приведено разъяснение смысла множества терминов, связанных с информатикой. Особое внимание уделено расшифровке замысловатых аббревиатур.

А.А. Зайцев.
Повесть
о настоящей
черепашке



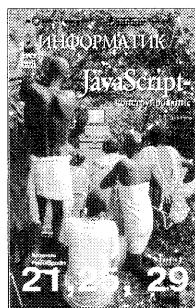
Этот увлекательный номер посвящен необычной (сказочной ©) методике изучения языка ЛОГО.

Д.М. Златопольский.
Внеклассная
работа
по информатике.
Сборник заданий
с решениями



В этот номер вошли разнообразные задания по информатике, которые можно использовать при проведении конкурсов, викторин и других внеклассных мероприятий с детьми различного возраста. Некоторые задания и занимательные игры можно использовать и на уроках.

ЖАРКОЕ ЛЕТО-2001



А.А. Дуванов. JavaScript-конструирование
Очередной учебник Роботландского университета. Курс "JS-конструирование" продолжает "HTML-конструирование" (см. № 21, 22/2000).



Л.Н. Картвелишвили.
На стенд в кабинете информатики

Целый номер "на стенд"? Да, целый номер! Причем весь этот номер был составлен из совершенно новых и очень интересных материалов. Первого сентября стенды в кабинетах наших читателей были лучшими в школе!



В.Н. Пинаев.
Олимпиады по информатике. Теория и практика

Даже если вы не очень любите олимпиады по информатике, они – объективная реальность, данная нам в телефонограммах. Как подготовить ребят к олимпиадам? Где взять задачи для проведения школьных олимпиад? Как эти олимпиады проводить? В номере на эти вопросы отвечает знающий человек.

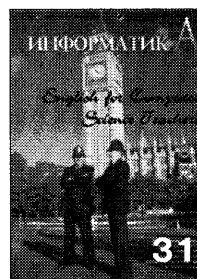


Иванова Н.Г., Русакова О.Л.
Информатика для школьников
5–6-х классов: ЛОГО-логика

Предлагаем вниманию подписчиков очень интересную разработку, созданную пермскими авторами, – рабочие тетради для параллельного изучения основ логики и языка ЛОГО.



С.М. Окулов. Основы программирования
В течение 2001 года в "Информатике" публиковались материалы (занятия) из новой книги нашего постоянного автора. В летних номерах мы завершили публикацию этой книги.



Л.Н. Картвелишвили.
English for Computer Science Teachers

How does a mouse work? Why aren't floppy disks floppy? How fast is the fastest computer? Have computers got brain? Where do viruses come from? Учителю информатики постоянно приходится иметь дело с англоязычными текстами. Достаточно ли уверенно вы чувствуете себя в английском? Конечно, один номер не может служить настоящим учебником, справочником или словарем. Но, мы надеемся, этот необычный номер стал полезен и интересен нашим читателям.



А.Г. Гейн.
Обязательный минимум содержания образования по информатике:

и в нем нам хочется дойти до самой сути
Что стоит за разделами минимума? Что имели в виду его авторы? (Такие вопросы тоже нередко возникают.) Давайте прочитаем минимум еще раз. Вместе, спокойно, не торопясь.



С.Л. Островский. Компьютерные вирусы

Наши постоянные подписчики, конечно, помнят, что более трех лет назад мы опубликовали последнюю редакцию «Компьютерных вирусов». С тех пор изменилось... все! Теперь мы предложили вниманию читателей совершенно новую книгу, интересную всем – и учителям, и ученикам, и даже их родителям.

ЖАРКОЕ ЛЕТО-2000

А.А. Дуванов.
HTML-
конструирование



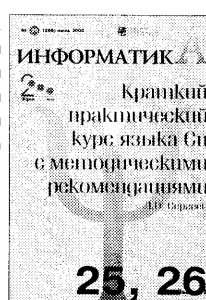
Пеци Р. Масани.
Норберт Винер:
жизнь
математика.
(Перевод
с английского
В.Гельман.)



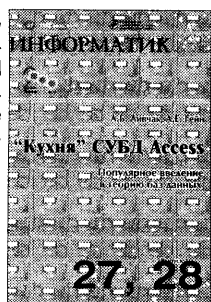
В.М. Нечев.
Локальная сеть
компьютерного
класса.
Вопросы
практического
использования



Л.О. Сергеев.
Краткий
практический
курс языка Си
с методическими
рекомендациями



А.Б. Ливчак,
А.Г. Гейн.
"Кухня" СУБД
Access.
Популярное
введение
в теорию
баз данных



Л.О. Сергеев.
Удивительный
мир языка Лисп.
Введение в язык
и задачи,
задачи, задачи...



Н.Л. Беленькая,
Л.О. Сергеев.
Фракталы



О.Л. Русакова.
Информатика:
уроки развития.
Материалы
для занятий
с учениками
начальной
школы



ЖАРКОЕ ЛЕТО-1999

А.А. Дуванов.
Интернет для
начинающих



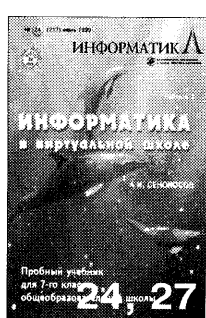
С.М. Окулов
и др.
Практикум
по Turbo
Паскалю



А.И. Маргулев.
Программирование
на языке
Visual Basic 5



А.И. Сенокосов.
Информатика
в виртуальной
школе.
Пробный учебник
для 7-го класса
общеобразовательной
школы



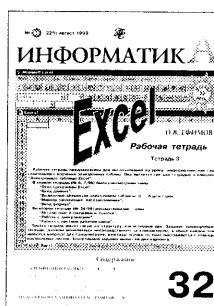
В.М. Нечев.
Модернизация
компьютерного
класса



Очерки
истории
информатики
в России



О.В. Ефимова.
Excel.
Рабочая
тетрадь



Вопросы, задания и контрольные работы для начинающих программистов

В.П. Гладков, А.П. Шестаков,

г. Пермь

Продолжение. См. № 20, 33, 34, 35, 37, 38, 40, 47/2001

Приведем 6 вариантов контрольных работ по материалам глав 1—8. Еще 6 вариантов контрольных работ будут опубликованы в № 5/2002.

Вариант 1

1. Изобразить на плоскости XOY область, в которой и только в которой истинно указанное выражение.

$\text{not}((y < -x + 1) \text{ and } (y < x + 1) \text{ and } (y > -x - 1) \text{ and } (y > x - 1)) \text{ and } (\text{sqr}(x) + \text{sqr}(y) \leq 1)$

2. Объяснить, в чем заключается синтаксическая ошибка (или ошибки) в приведенной программе. Исправить эти ошибки.

```
program ошибки;
var x, y: integer;
begin
  writeln(введите значение X);
  readln(x);
  writeln(введите значение Y);
  readln(y);
  s := x + y;
  if s < 0 then s := abs(s)
  writeln(s)
end.
```

3. Решить задачу с использованием условного оператора.

Составить программу для определения вида треугольника по двум углам α_1 и α_2 . Определяемое свойство: остроугольный, прямоугольный, тупоугольный (для корректного анализа не надо забывать о величине третьего угла).

4. Решить задачу с использованием оператора выбора.

По введенному номеру типа вагона определить стоимость проезда в поезде (1 — общий; 2 — плацкартный; 3 — купейный; 4 — СВ). Стоимость проезда в общем вагоне — S руб., с повышением комфортности она увеличивается в каждом случае в 1,5 раза по сравнению с предыдущим.

5. Точно и четко сформулировать условие задачи, решение которой приведено в данной программе.

```
program pr;
var k, i, d, s: integer;
begin
  writeln('введите натуральное число n: ');
  readln(n);
  writeln('введите значение k: ');
  readln(k);
  d := 2; i := 0;
```

```
while d <= n div 2 + 1 do
begin if (n mod d = 0) and (d > k)
then i := i + 1;
d := d + 1
end;
writeln(i)
end.
```

6. Решить задачу с использованием цикла.

Найти сумму N простых чисел в арифметической прогрессии 11, 13, 15, 17, 19, 21, ...

7. Выписать последовательно значения, которые будут выводиться программой по ходу ее выполнения.

```
program schet;
var i, j: word;
begin i := 0; j := 0;
  repeat i := i + 2;
    j := j + 3;
    write(i:5, j:3)
  until i + j > 20;
  writeln;
  i := 0; j := 0;
  while i + j < 20 do
  begin i := i + 2;
    j := j + 3;
    write(i:5, j:3)
  end
end.
```

8. Решить задачу с использованием подпрограммы.

Найти все четырехзначные числа, у которых сумма крайних цифр равна сумме средних цифр, а само число делится на 6 и 27.

9. Решить задачу.

В данной последовательности $a_1, a_2, \dots, a_n$ определить минимальное число среди элементов с четными номерами. Например, для последовательности —1, 0, 12, —77, 22, —6, 70, 11, 3 получаем ответ —77.

10. Решить задачу.

Вычислить значение выражения

$$\sqrt{3 + \sqrt{6 + \dots + \sqrt{96 + \sqrt{99}}}}$$

Вариант 2

1. Изобразить на плоскости XOY область, в которой и только в которой истинно указанное выражение.

$$(\text{sqr}(x) + \text{sqr}(y) \leq 1) \text{ and } \\ (y \geq x) \text{ and } (y \geq -x)$$

2. Объяснить, в чем заключается синтаксическая ошибка (или ошибки) в приведенной программе. Исправить эти ошибки.

```
program д;
var x: real;
begin
  writeln(введите значение x);
  readln(x);
  y := sqr(x) - 1;
  if y < 0 then y := y * -1
  writeln(y);
end
```

3. Решить задачу с использованием условного оператора.

Составить программу для определения вида треугольника по сторонам a , b и углу α между ними. Определяемое свойство: равносторонний, равнобедренный или прямоугольный (обратите внимание, что треугольник может быть равнобедренным прямоугольным). Для корректного анализа нужно определить еще третью сторону (например, по теореме косинусов $c^2 = a^2 + b^2 - 2ab \cdot \cos \alpha$).

4. Решить задачу с использованием оператора выбора.

По введенной дате рождения (число и месяц) определить знак зодиака, под которым родился человек.

5. Точно и четко сформулировать условие задачи, решение которой приведено в данной программе.

```
program pr;
var k, i, d, s: integer;
begin s := 0;
  for i := 1 to 200 do
    begin d := 2;
      k := 0;
      while d <= i div 2 + 1 do
```

```
begin if i mod d = 0 then k := k + 1;
      d := d + 1
    end;
    if k = 5 then s := s + i;
  end
end.
```

6. Решить задачу с использованием цикла.

Найти k -е число в геометрической прогрессии 2, 8, 32, 128, 512, ..., которое кратно m или n .

7. Выписать последовательно значения, которые будут выводиться программой по ходу ее выполнения.

```
program schet;
var i, j: word;
begin i := 0; j := 0;
  repeat i := i + (i + 1);
    j := j + 1;
    write(i:5, j:3)
  until i + j > 20;
  writeln;
  i := 0; j := 0;
  while i + j < 18 do
    begin
      i := i + (i + 1);
      j := j + 1;
      write(i:5, j:3)
    end
  end.
```

8. Решить задачу с использованием подпрограммы.

Для каждой тройки, которую можно составить из отрезков a , b , c , d , определить, можно ли из них построить треугольник, и если да, то найти его площадь.

9. Решить задачу.

В данной последовательности $a_1, a_2, \dots, a_n$ определить максимальное и минимальное числа среди элементов с нечетными номерами. Например, для последовательности $-1, 0, 12, -77, 22, -6, 70, 11, 3$ получаем ответы 70 и -1 .

10. Решить задачу.

Вычислить значение выражения

$$\cos(1 + \cos(2 + \dots + \cos(39 + \cos 40) \dots)).$$

Вариант 3

1. Изобразить на плоскости XOY область, в которой и только в которой истинно указанное выражение.

$(x \geq -1) \text{ and } (x \leq 0) \text{ and } (y \geq -2) \text{ and } (y \leq -x) \text{ or } (x \geq 0) \text{ and } (x \leq 1) \text{ and } (y \geq -2) \text{ and } (y \leq x)$

2. Объяснить, в чем заключается синтаксическая ошибка (или ошибки) в приведенной программе. Исправить эти ошибки.

```
program summa;
begin
  var x, y: integer;
      z: boolean;
  writeln(' введите x и y ');
  readln(x; y);
  if x + y > 0 then z := true;
  else z := false;
  writeln(z)
end.
```

3. Решить задачу с использованием условного оператора.

Составить программу для определения вида четырехугольника с двумя параллельными сторонами по двум прилежащим углам α_1 и α_2 . Определяемое свойство: обыкновенная трапеция, прямоугольная трапеция, равнобедренная трапеция, прямоугольник.

4. Решить задачу с использованием оператора выбора.

Для заданного натурального числа N определить, на какую цифру будет оканчиваться число, полученное из данного умножением на два.

5. Точно и четко сформулировать условие задачи, решение которой приведено в данной программе.

```
program pr;
var k, i, d, s, n: integer;
begin
  writeln('введите натуральное число n: ');
  readln(n);
  writeln('введите значение k: ');
  readln(k);
  d := 3; i := 0;
```

```
while d <= n div 2 + 1 do
begin if (n mod d = 0) and (d > k)
then i := i + 1;
d := d + 2
end;
writeln(i)
end.
```

6. Решить задачу с использованием цикла.

Найти k -е число в геометрической прогрессии 3, 9, 27, 81, 243, ...

7. Выписать последовательно значения, которые будут выводиться программой по ходу ее выполнения.

```
program schet;
var i, j: word;
begin i := 0; j := 0;
  repeat j := j + 1; i := i + (j - i);
  write(i:4, j:2)
  until j mod 5 = 0;
  writeln;
  i := 0; j := 0;
  while j < 4 do
  begin
    j := j + 1;
    i := i + (j - i);
    write(i:4, j:2)
  end
end.
```

8. Решить задачу с использованием подпрограммы.

По координатам вершин двух треугольников определить, какой из них имеет большую площадь.

9. Решить задачу.

В данной последовательности $a_1, a_2, \dots, a_n$ определить максимальное и минимальное числа среди элементов с четными номерами. Например, для последовательности $-1, 0, 12, -77, 22, -6, 70, 11, 3$ получаем ответы 11 и -77 .

10. Решить задачу.

Вычислить

$$(1 + 2)(1 + 2 + 3) \cdot \dots \cdot (1 + 2 + \dots + N).$$

Вариант 4

1. Изобразить на плоскости XOY область, в которой и только в которой истинно указанное выражение.

$$(\text{sqr}(x) + \text{sqr}(y - 1) \leq 1) \text{ and } (y \leq 1 - \text{sqr}(x))$$

2. Объяснить, в чем заключается синтаксическая ошибка (или ошибки) в приведенной программе. Исправить эти ошибки.

```
program max
var a, b, c: real;
begin
  writeln(' введите три числа');
  readln(a, b, c);
  if a > b then b := a;
  else b := b;
  if b > c then writeln(b);
  else writeln(c)
end.
```

3. Решить задачу с использованием условного оператора.

Составить программу для определения вида параллелограмма по прилежащим сторонам a , b и углу между ними α . Определяемое свойство: ромб, прямоугольник, квадрат или обыкновенный параллелограмм.

4. Решить задачу с использованием оператора выбора.

В некоторой компьютерной игре в зависимости от уровня сложности игроку присваивается тот или иной титул (I — атаман; II — главнокомандующий; III — принц; IV — король; V — император; VI — божество). По введенному номеру уровня сложности указать соответствующий титул.

5. Точно и четко сформулировать условие задачи, решение которой приведено в данной программе.

```
program pr;
var k, m, i, d, a, b: integer;
begin
  writeln('введите a и b: ');
  readln(a, b);
  writeln('введите значение m: ');
  readln(m);
  for i := a to b do
    begin
      d := 2;
```

```
k := 0;
while d <= i div 2 + 1 do
begin
  if i mod d = 0 then k := k + 1;
  d := d + 1
end;
if k > m then writeln(i)
end
end.
```

6. Решить задачу с использованием цикла.

Найти сумму k чисел, стоящих на четных местах в геометрической прогрессии 2, 8, 32, 128, 512, ...

7. Выписать последовательно значения, которые будут выводиться программой по ходу ее выполнения.

```
program schet;
var i, j: word;
begin i := 0; j := 0;
  repeat i := i + 1; j := j + (i - j);
    write(i:4, j:2)
  until i > 5;
  writeln;
  i := 0; j := 0;
  while i < 5 do
    begin
      i := i + 1;
      j := j + (i - j);
      write(i:4, j:2)
    end
  end.
```

8. Решить задачу с использованием подпрограммы.

Подсчитать число всех натуральных чисел, меньших M , квадрат суммы цифр которых равен X .

9. Решить задачу.

Даны натуральное число n и вещественные числа $a_1, a_2, \dots, a_n$. Определить, есть ли в последовательности числа, которые кратны трем. В случае положительного ответа определить порядковый номер первого из них.

10. Решить задачу.

Даны числа x и a . Вычислить значение выражения $((\dots((x + a)^2 + a)^2 + \dots + a)^2 + a)^2 + a$ (всего n скобок).

Вариант 5

1. Изобразить на плоскости XOY область, в которой и только в которой истинно указанное выражение.

$$(y \geq 1) \text{ or } (y \leq 1) \text{ and } (y \geq 0) \text{ and } (y \geq x) \text{ and } (y \geq -x)$$

2. Объяснить, в чем заключается синтаксическая ошибка (или ошибки) в приведенной программе. Исправить эти ошибки.

```
program myerror;
var a, b, min;
begin
  writeln(' введите a и b');
  readln(a, b);
  if a < b then min := a;
  writeln('да. ');
  else min := b;
  writeln('нет. ');
  writeln(min:5:2)
end.
```

3. Решить задачу с использованием условного оператора.

Составить программу для определения вида четырехугольника, у которого две противоположные стороны параллельны, а две другие — равны. Определить по двум прилежащим сторонам a , b и углу α между ними вид четырехугольника: равнобедренная трапеция, прямоугольник, квадрат.

4. Решить задачу с использованием оператора выбора.

Составить программу, которая, если введенный символ является заглавной русской буквой, выводит женское имя, начинающееся с этой буквы, иначе сообщает об ошибочном вводе.

5. Точно и четко сформулировать условие задачи, решение которой приведено в данной программе.

```
program pr;
var s: longint;
    a, b, c: 0..9; i: 100..999;
begin s := 0;
  for i := 100 to 999 do
    begin a := i div 100;
          b := i div 10 mod 10;
          c := i mod 10;
          if (a mod 2 = 1) and (b mod 2 = 1) and
             (c mod 2 = 1)
          then s := s + i
```

```
end;
  writeln(s)
end.
```

6. Решить задачу с использованием цикла.

Найти все целые корни уравнения

$$ax^3 + bx^2 + cx + d = 0,$$

где a, b, c, d — заданные целые числа, причем $a \neq 0$ и $d \neq 0$.

Замечание: целыми корнями могут быть только положительные и отрицательные делители коэффициента d .

7. Выписать последовательно значения, которые будут выводиться программой по ходу ее выполнения.

```
program schet;
var i, j: word;
begin i := 0; j := 0;
  repeat i := i + 1; j := j + (i + 10);
    write(i:5, j:3)
  until j > 50;
  writeln;
  i := 0; j := 0;
  while j < 40 do
    begin
      i := i + 1;
      j := j + (i + 10);
      write(i:5, j:3)
    end
  end.
```

8. Решить задачу с использованием подпрограммы. Вычислить $z = (\text{sign } x + \text{sign } y) \cdot \text{sign}(x + y)$, используя для этого функцию вычисления $\text{sign } \alpha$ по правилу

$$\text{sign } \alpha = \begin{cases} 1, & \text{если } \alpha > 0, \\ 0, & \text{если } \alpha = 0, \\ -1, & \text{если } \alpha < 0, \end{cases}$$

9. Решить задачу.

Даны натуральное число n и вещественные числа $a_1, a_2, \dots, a_n$, среди которых есть хотя бы одно отрицательное. Найти величину наибольшего среди отрицательных чисел последовательности.

10. Решить задачу.

Вычислить значение выражения

$$\sqrt{3 + \sqrt{6 + \dots + \sqrt{3 \cdot (n-1) + \sqrt{3n}}}}$$

Вариант 6

1. Изобразить на плоскости XOY область, в которой и только в которой истинно указанное выражение.

$$(y \leq -x + 1) \text{ and } (y \geq -x - 1) \text{ and } (\text{sqr}(x) + \text{sqr}(y) \leq 1)$$

2. Объяснить, в чем заключается синтаксическая ошибка (или ошибки) в приведенной программе. Исправить эти ошибки.

```
program myerror
var a, x: real;
begin
  write(x);
  readln(x);
  if x >= 0 then a:=x;
  else a:=|x|;
  writeln(a)
end.
```

3. Решить задачу с использованием условного оператора.

Составить программу для определения вида ромба по углу α . Определяемое свойство: ромб или квадрат.

4. Решить задачу с использованием оператора выбора.

Имеются семь матрешек, вложенных друг в друга. По заданному номеру матрешки вывести ее высоту, если известно, что самая большая из них имеет высоту H , а каждая последующая имеет высоту, отличающуюся от предыдущей на L .

5. Точно и четко сформулировать условие задачи, решение которой приведено в данной программе.

```
program pr;
var s, m, i, d, a, b: integer;
begin
  writeln('введите a и b: ');
  readln(a,b);
  writeln('введите значение m: ');
  readln(m);
  for i := a to b do
  begin d := 3;
    s := 0;
    while d <= i div 2 + 1 do
```

```
begin if i mod d = 0 then s := s + 1;
      d := d + 2
    end;
    if k = m then writeln(i)
  end.
```

6. Решить задачу с использованием цикла.

Найти наибольшее и наименьшее значения функции $y = 3x^2 + x - 4$, если на заданном интервале $[a, b]$ значение x изменяется с шагом 0,1.

7. Выписать последовательно значения, которые будут выводиться программой по ходу ее выполнения.

```
program schet;
var i, j: word;
begin i := 0; j := 0;
  repeat i := i + 2; j := j + (i + 12);
    write(i:5, j:3)
  until j > 54;
  writeln;
  i := 0; j := 0;
  while j < 40 do
  begin
    i := i + 2;
    j := j + (i + 12);
    write(i:5, j:3)
  end
end.
```

8. Решить задачу с использованием подпрограммы. Составить программу подсчета количества всех натуральных чисел, меньших заданного M , квадрат суммы цифр которых делится на X .

9. Решить задачу.

Даны натуральное число n и целые трехзначные числа $a_1, a_2, \dots, a_n$. Определить, есть ли в последовательности числа, в записи которых используется цифра 9. В случае положительного ответа определить порядковый номер первого из них.

10. Решить задачу.

Вычислить значение выражения $\sin(1 + \sin(2 + \dots + \sin(19 + \sin 20) \dots))$.

Уильям Бэрроуз



Изобретение клавишного ввода чисел (взамен медленного ручного) для счетных машин [1—6] позволило в середине 80-х годов XIX столетия наладить их промышленный выпуск. Свой вклад в создание клавишных машин внесли изобретатели многих стран, но одно из главных мест занимает здесь американец Уильям Бэрроуз (1857—1898).

Описание его жизни — хороший материал для Голливуда [1]: в ней были и голодное детство, и утомительный труд, и тяжелая болезнь, и мечта, и напряженная работа ради ее осуществления, связанная с лишениями, и, наконец, успех, слава и богатство (к сожалению, слишком поздние).

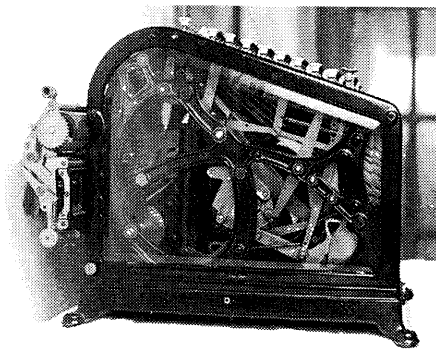
Бэрроуз родился в городке Рочестер (штат Нью-Йорк). Его отец в поисках заработка скитался с семьей по Америке, пока не осел в другом городе того же штата — Оберне. Здесь Уильям стал посещать начальную школу, но вскоре был отдан учеником бухгалтера в местный банк, где и остался потом работать. Ежедневное многочасовое сидение над колонками цифр в маленьких душных комнатах не способствовало улучшению и без того не очень крепкого здоровья Уильяма Бэрроуза. Он заболел туберкулезом и, оставив по совету врача бухгалтерскую работу, в 1882 году переехал в Сент-Луис, где поступил в ремонтную мастерскую на должность механика.

Бэрроуз хорошо понимал, что изготовление машин, которые облегчают труд, связанный с однообразными утомительными вычислениями, может стать прибыльным делом. Он заинтересовал будущими прибылями хозяина мастерской Д.Бойера, а также местного фабриканта Т.Меткалфа и, получив от них 700 долларов, начал трудиться. Это было в 1884 году. Однако денег хватило ненадолго — материалы и инструменты стоили больше, чем предполагал Бэрроуз, да к тому же первая модель машины оказалась неудачной. Деньги на новую модель ему пришлось просить у другого человека — предпринимателя Р.М. Скраггса. Однако и вторая модель не оправдала ожиданий. Бэрроуз строит третью мо-

дель и, поскольку ему кажется, что на этот раз все получилось как надо, делает сразу несколько экземпляров машины. Но и тут его постигло разочарование: попытки обучить других людей обращаться с машиной терпели неудачу — слишком сильный или слишком слабый удар по клавишам нарушал ее нормальную работу.

Тем не менее изобретатель не терял уверенности в конечном успехе. В конце 1885 года он закончил постройку новой модели, а в январе следующего года У.Бэрроуз, Т.Меткалф, Р.М. Скраггс и еще один местный предприниматель Х.Пай создают Американскую компанию арифмометров — одну из первых в мире фирм, занимающихся производством счетных машин.

[Машина Бэрроуза являлась двухтактной: на первом такте осуществлялась установка числа с помощью клавиш, а на втором — установлен-



Машина Бэрроуза

ное число движением приводного рычага переносилось на счетчик. Таким образом, клавиши не имели отношения к действиям, выполняемым самой машиной, и оставались в опущенном положении с момента установки числа. Это позволяло непосредственно осуществлять контроль ввода и в случае необходимости исправлять ошибки. Машина могла печатать исходные числа, производить сложение (или вычитание) и печатать результат. Использовать ее для выполнения умножения было трудно.

В дальнейшем неоднократно предпринимались попытки усовершенствовать машину Бэрроуза. Удалось, например, расширить перечень выполняемых на ней операций, в частности, появились операции “Печать без сложения”, “Сложение без печати”, “Поперечное сложение”, “Печать списков и таблиц”. Впоследствии приводной рычаг был заменен электрическим двигателем.]

Дела у новой компании пошли так успешно, что вскоре Бэрроуз превратился из бедняка в состоятельного бизнесмена. Но жить ему оставалось уже недолго — 14 сентября 1898 года Уильям Бэрроуз скончался. На его могиле написано [1]: “Здесь покоится человек, который был благородным в бедности, скромным в богатстве и великим в своих делах на благо человечества”.

В XX веке корпорация “Бэрроуз” (Burroughs) стала одним из крупнейших в мире производителей компьютеров. В частности, этой фирмой выпущен первый компьютер на интегральных схемах (1968 г.), созданы многопроцессорные системы B5500 (1962 г.), B6500, B6700, B7700, B-3800 (1976 г.), B-4800 (1976 г.), B-6800 (1977 г.), B-7800 (1978 г.), где были воплощены многие новые идеи параллельной обработки данных, и система B-7900 (1982—1983 гг.), в основу архитектуры которой положен принцип так называемой распределенной обработки (предполагающий использование нескольких специализированных функциональных подсистем различного назначения и разной производительности) [7].

Литература

1. Гутер Р.С., Полунов Ю.Л. От абака до компьютера. Изд. 2-е, испр. и доп. М.: Знание, 1981.
2. Вычислительные машины // Большая советская энциклопедия. Изд. 2-е. М.: Гл. науч. изд-во “Большая советская энциклопедия”, 1954. Т. 9.
3. Знакомьтесь: компьютер: Пер. с англ. М.: Мир, 1989.
4. Печерский Ю.Н. Этюды о компьютерах. Кишинев: Штиница, 1989.
5. Леонов А.Г., Четвергова О.В. История компьютеров // Информатика, № 35/98.
6. Механические калькуляторы // Информатика, № 26/2001.
7. Частиков А.П. От калькулятора до суперЭВМ // Новое в жизни, науке, технике. Сер. “Вычислительная техника и ее применение”, № 1/88.

Читайте в номере

Страницы повышения квалификации 2-7

Е.В. Андреева. Олимпиады по информатике. Пути к вершине

“Задачи дискретной математики, к которым относится большинство олимпиадных задач по информатике, часто сводятся к перебору различных комбинаторных конфигураций объектов и выбору среди них наилучшего с точки зрения условий той или иной задачи”.
Лекция № 6 целиком посвящена комбинаторным задачам.

Книжный шкаф учителя информатики 8

Книга “Алексей Андреевич Ляпунов” из серии “История информатики”

“...в интересах личности и общества в целом нецелесообразно “стричь всех под одну гребенку”, предъявлять ко всем одинаковые требования и вести всех по единой педагогической стезе”.
О книге, посвященной человеку, который сыграл решающую роль в борьбе за признание и развитие кибернетики в нашей стране.

Газете “Информатика” 7 лет 9-12, 21-24

Наша газета уже в течение семи лет каждую неделю приходит в ваш дом или школу

В этом последнем номере уходящего 2001 года мы напоминаем вам, дорогие читатели, об основных рубриках газеты, о новых проектах и о традиционных выпусках “Жаркое лето”.

Тематический указатель материалов за 2001 год 13-20

Тематический указатель материалов, опубликованных в газете “Информатика” в 2001 году (№ 1-48)

Материалы к уроку 25-30

В.П. Гладков, А.П. Шестаков. Вопросы, задания и контрольные работы для начинающих программистов

Могут ли ваши ученики составить программу (с использованием оператора цикла) для нахождения суммы k чисел, стоящих на четных местах в геометрической прогрессии 2, 8, 32, 128, 512, ...?
Продолжение публикации (начатой в № 20, 33-35, 37, 38, 40, 47/2001), цель которой — помочь начинающему программисту освоить язык программирования Паскаль. В этом номере представлены шесть вариантов контрольных работ.

На стенд в кабинете информатики 31

Уильям Бэрроуз

О человеке, именем которого названа одна из крупнейших в XX столетии фирм — производителей компьютеров.

Гл. редактор С.Л. Островский Зам. гл. редактора А.И. Сенокосов Редакция: Е.В. Андреева Н.Л. Беленькая Л.Н. Картвелишвили Н.П. Медведева Дизайн и верстка: Н.И. Пронская Корректоры: Е.Л. Володина, С.М. Подберезина	©ИНФОРМАТИКА 2001 выходит четыре раза в месяц При перепечатке ссылка на ИНФОРМАТИКУ обязательна, рукописи не возвращаются	Адрес редакции и издателя: 121165, Киевская, 24 тел. 249-48-96 Отдел рекламы тел. 249-98-70	Учредитель: ООО “Чистые пруды” Зарегистрировано в Министерстве РФ по делам печати. ПИ № 77-7230 от 12.04.2001. Отпечатано в ОИД “Медиа-Пресса”, 125993, ГСП-3, Москва, А-40, ул. “Правды”, 24. Тираж 6000 экз. Срок подписания в печать по графику 12.12.2001. Номер подписан 12.12.2001. Заказ № 15218 Цена свободная
ИНДЕКС ПОДПИСКИ для индивидуальных подписчиков 32291 комплекта изданий 32744			
Тел.: (095)249-31-38, 249-33-86. Факс (095)249-31-84			
Internet: inf@1september.ru WWW: http://www.1september.ru			

Главный редактор комплекта изданий А.С. Соловейчик

Английский язык (Е.В. Громушкина), индекс подписки — 32025; **Библиотека в школе** (О.К. Громова), индекс подписки — 33376; **Биология** (Н.Г. Иванова), индекс подписки — 32026; **Воскресная школа** (монах Киприан (Яценко), индекс подписки — 32742; **География** (О.Н. Коротова), индекс подписки — 32027; **Дошкольное образование** (М.С. Аромштам), индекс подписки — 33373; **Здоровье детей** (А.У. Лекманов), индекс подписки — 32033; **Информатика** (С.Л. Островский), индекс подписки — 32291; **Искусство** (Н.Х. Исмаилова), индекс подписки — 32584; **История** (А.Ю. Головатенко), индекс подписки — 32028; **Литература** (Г.Г. Красухин), индекс подписки — 32029; **Математика** (И.Л. Соловейчик), индекс подписки — 32030; **Начальная школа** (М.В. Соловейчик), индекс подписки — 32031; **Немецкий язык** (М.Д. Бузоева), индекс подписки — 32292; **Русский язык** (Л.А. Гончар), индекс подписки — 32383; **Спорт в школе** (Н.В. Школьникова), индекс подписки — 32384; **Управление школой** (А.И. Адамский), индекс подписки — 32652; **Физика** (Н.Д. Козлова), индекс подписки — 32032; **Французский язык** (Г.А. Чесновицкая), индекс подписки — 33371; **Химия** (О.Г. Блохина), индекс подписки — 32034; **Школьный психолог** (М.Н. Сартан), индекс подписки — 32898.