

УДК 681.3.06

РАБОТА С ИЗОБРАЖЕНИЯМИ В ГЛАВНОМ УПРАВЛЯЮЩЕМ ЯЗЫКЕ БАЗОВОЙ ТЕХНОЛОГИИ СИСТЕМ ОБРАБОТКИ И ОТОБРАЖЕНИЯ СЛОЖНО ОРГАНИЗОВАННОЙ ИНФОРМАЦИИ

И. В. Зюряева¹, Т. В. Ламзина¹

Рассматриваются вопросы обработки и хранения графической информации, возникающие при разработке систем визуализации данных. Обсуждаются форматы хранения изображений и способы установки текущей палитры экрана. Предложены приемы работы с изображениями в зависимости от спецификации координат источника и адресата.

Ключевые слова: визуализация, графическая информация, палитра экрана, сжатие графической информации, инструментальные программные средства.

1. Введение. В настоящее время при разработке систем визуализации данных все большее значение приобретают вопросы обработки и хранения графической информации. Существует большое количество различных форматов файлов, используемых для работы с графикой (см. [1, 2, 5]). Эти возможности отражают возможности аппаратной видеосистемы компьютера, различные способы сжатия графической информации, а также функциональные назначения графических изображений.

Главный управляющий язык Базовой технологии создания систем обработки и отображения сложно организованной информации (см.[3]), разрабатываемой в НИВЦ МГУ, оперирует с фиксированными видами графических форматов файлов. Средствами этого языка генерируются, преобразовываются и отображаются на экран дисплея изображения в требуемых форматах. Данная статья посвящена анализу основных понятий и описанию работы с изображениями на Главном управляющем языке.

2. Общие понятия, связанные с изображениями и видеорежимами. Изображение на экране определяется совокупностью пикселей (экранных точек), каждый из которых имеет свой цвет. Цвет пикселя образуется из сочетания трех компонент различной интенсивности: Red — красный, Green — зеленый, Blue — синий (так называемые RGB-компоненты). Интенсивность каждой компоненты может меняться в пределах от 0 до 255, причем RGB=(0,0,0) соответствует черному цвету, а RGB=(255,255,255) — белому. Основными параметрами изображения являются глубина цвета и разрешение. В общем случае глубина цвета определяется количеством цветов, которые можно использовать в изображении, а разрешение — количеством пикселей изображения, выводимых на экран.

Все видеорежимы по способу задания цветовой характеристики изображения разделяются на два класса. В первом классе видеорежимов задание пикселя изображения осуществляется путем ссылки на элемент палитры, содержащей соответствующие значения RGB. Палитра экрана может содержать 2, 16 или 256 цветов, что определяется количеством разрядов (битов), отводимых для задания каждого пикселя изображения (1, 4 или 8 битов). Во втором классе видеорежимов каждый пиксель изображения задается путем явного указания RGB-компонент, т.е. без использования палитры. При этом количество порождаемых цветов зависит от количества битов, отводимых для представления RGB-компонент и может содержать 15 или 16 битов, что обеспечивает 32 или 64 тысячи цветов (режим High Color), либо 24 или 32 бита, что обеспечивает 16 миллионов цветов (режим True Color).

Главный управляющий язык позволяет осуществлять установку видеорежима в пределах режимов, поддерживаемых видеоаппаратурой, или формулировать запрос пользователя на выдачу информации о текущем режиме (команда screen_mode). Номера поддерживаемых режимов могут быть получены с помощью системной программы vesainfo.viz. Номера и характеристики существующих видеорежимов приведены в табл. 1.

3. Форматы изображений. При записи экранного изображения в файл создается матрица пикселей (т.е. матрица, описывающая цветовую характеристику каждого пикселя изображения), которая соответствует по своим размерам ширине и высоте изображения на экране в пикселях. Эта матрица называется растром. Способы задания такого раstra разнообразны и определяются различными форматами файлов, допустимыми в Главном управляющем языке: VSE, CEL, BMP, PCX, GIF, TIF, JPG. Форматы отличаются разными способами хранения изображения. Существует два способа задания цветов.

¹ Научно-исследовательский вычислительный центр, Московский государственный университет им. М.В. Ломоносова, 119899, Москва; e-mail: iraz@srcc.msu.su

1) Задание цветов изображения (RGB-компонент) в растре при помощи палитры, которая хранится в файле вместе с растром. Используются палитры, содержащие 2 (1 бит/пиксель), 16 (4 бит/пиксель) и 256 (8 бит/пиксель) цветов.

2) Непосредственное задание цветов изображения, когда каждая RGB-компонента описывается в самом растре (без использования палитры).

Таблица 1

Номер видеорежима	Размер экрана (пикс.)	Количество цветов	Название режима
10	640x350	16	EGA
11	640x480	2	MONO
12	640x480	16	VGA
13	320x240	256	VGA
100	640x400	256	VESA
101	640x480	256	VESA
102	800x600	16	VESA
103	800x600	256	VESA
104	1024x768	16	VESA
105	1024x768	256	VESA
106	1280x1024	16	VESA
107	1280x1024	256	VESA
110	640x480	32K	VESA
111	640x480	64K	VESA
11A	1280x1024	64K	VESA
11B	1280x1024	16M	VESA
112	640x480	16M	VESA
113	800x600	32K	VESA
114	800x600	64K	VESA
115	800x600	16M	VESA
116	1024x768	32K	VESA
117	1024x768	64K	VESA
118	1024x768	16M	VESA
119	1280x1024	32K	VESA
120	1600x1200	256	VESA
121	1600x1200	32K	VESA
122	800x600	16	VESA
124	1024x768	16	VESA
128	1280x1024	16	VESA
136	1152x864	256	VESA

При способе задания 15 бит/пиксель или 16 бит/пиксель под компоненту отводится по 5 пикселей (во втором случае для компоненты G отводится 6 пикселей), что обеспечивает задание 32K или 64K цветов (High Color). При способе задания 24 бит/пиксель или 32 бит/пиксель под компоненту отводится по 1 байту (во втором случае для удобства доступа добавляется один неиспользуемый байт), что обеспечивает задание 16M цветов (True Color).

Вторым элементом, характеризующим растр, является количество плоскостей, определяющих число матриц хранения элементов раstra. Используются одна, три и четыре плоскости:

- одна плоскость: полный растр изображения состоит из одной матрицы;
- три плоскости: полный растр изображения состоит из трех матриц для каждой RGB-компоненте;
- четыре плоскости: полный растр изображения состоит из четырех матриц, в которых каждый пиксель изображения хранится специальным образом (подробнее см. [4]).

Сочетания параметров, допустимые для хранения изображения при помощи различных форматов, приведены в табл. 2.

Используются следующие форматы хранения и сжатия изображений:

VSF — внутренний формат хранения, предназначенный для быстрой оперативной работы с изображениями;

CEL — внутренний формат хранения с разбивкой изображения на клетки; предназначен для обработки больших (десятки и сотни мегабайт) изображений;

BMP — формат использует простые методы сжатия, что делает его пригодным для быстрого чтения

и записи небольших изображений; широко используется в операционной системе WINDOWS;

PCX — один из первых появившихся форматов для хранения растровых палитровых изображений;

GIF — универсальный графический формат для хранения палитровых изображений, обеспечивающий значительное сжатие данных;

TIF — формат обеспечивает хранения графических данных в структурированном виде, что позволяет осуществлять быстрый доступ к различным фрагментам большого изображения; широко применяется для устройств сканирования изображения;

JPG — формат предназначен для работы с полноцветными (True Color) изображениями фотографического качества и использует эффективные методы сжатия.

Таблица 2.

Формат файла	Количество бит на пиксель	Количество плоскостей
BMP	1	1
	4	1
	8	1
	24	1
PCX	1	1
	1	4
	8	1
	8	3
GIF	8	1
JPG	24	1
VSF	1	1
VSF	1	4
VSF	4	1
VSF	8	1
VSF	15	1
VSF	16	1
VSF	24	1
VSF	32	1

4. Палитра и индексы. Текущая палитра экрана может быть изменена путем установки и занесения новых значений RGB-компонент для заданных номеров цветов палитры. Палитра представляет собой совокупность четырехбайтовых целых чисел. Первое число — количество цветов в палитре, далее следуют элементы палитры в формате 0BGR (старший байт — ноль, затем интенсивности синего, зеленого и красного цветов). Для удобства использования определены следующие виды палитр.

1) Серая палитра, состоящая из градаций серого цвета. Интенсивность всех RGB-компонент задаваемых номеров цветов является одинаковой в каждом номере цвета и определяется в интервале от 0 до 255 для номеров цветов в заданном диапазоне. Крайние номера цветов в диапазоне имеют интенсивность 0 и 255, а интенсивность остальных номеров цветов диапазона равномерно возрастает и зависит от количества номеров цветов в заданном диапазоне. Например, в диапазоне из четырех номеров цветов интенсивность всех RGB-компонент двух средних цветов составит 85 и 170. Если в диапазон входит только один номер цвета, то интенсивность всех его RGB-компонент полагается равной 128.

2) Черная палитра. Интенсивность всех RGB-компонент задаваемых номеров цветов равна нулю.

3) Белая палитра. Интенсивность всех RGB-компонент задаваемых номеров цветов равна 255.

4) Одноцветная палитра. Интенсивность всех RGB-компонент задаваемых номеров цветов совпадает.

Аппаратно устанавливаются стандартные значения палитры. Создание новой палитры и ее модификация производится в Главном управляющем языке посредством команд `palette_get3` и `palette_set3`. Основными параметрами команд являются палитра `pal`, записанная в формате 0BGR, и индексы `ind`, определяющие диапазоны цветов в палитре (диапазон — это подряд идущие номера цветов), для которых устанавливаются новые значения RGB-компонент из `pal`. Палитра определяет цвета, которые подлежат изменению, причем остальные цвета текущей палитры сохраняются.

Первое четырехбайтовое число в индексе `ind` представляет собой количество диапазонов номеров цветов. Далее следует описание этих диапазонов (по одному четырехбайтовому целому числу на каждый диапазон, в котором два младших байта определяют начальный (младший) номер цвета в диапазоне, а два старших байта — количество номеров цветов в этом диапазоне). Общее количество цветов в индексе должно совпадать с заданным количеством цветов в палитре.

Диапазоны цветов могут быть заданы непосредственно в переменной `pal` и располагаться после описания цветов палитры. Если задан индекс `ind`, то при получении требуемой палитры указанные диапазоны переносятся в конец переменной `pal`. Если диапазоны явно не определены, то считается, что в переменной `pal` должны присутствовать все номера цветов текущего видеорежима, а в конец переменной `pal` заносится диапазон в виде четырехбайтового нулевого числа.

Приведем примеры создания и модификации палитры.

```
* создание серой палитры для [9-10] [3-6] и 12-го цветов
bvar_set ind 0 "1D 6d" 3 9 2 3 4 12 1
palette_get3 2 b_pal ind
```

```
* установка новых значений для 0-го и 15-го элементов палитры:
* синий (1) - желтый (255 255 0),
* желтый (14) - синий (0 0 255)
bvar_set n_pal 0 "1D 8b 1D 4d" 2
          255 255 0 0 0 0 255 0 2 1 1 14 1
palette_set3 2 n_pal
```

5. Основные приемы работы с изображениями. Приведем описания параметров (указываются в фигурных скобках) и команд (указываются в квадратных скобках) для работы с изображениями в Главном управляющем языке.

При записи [`picture_save3`], показе [`picture_show3`] и преобразовании [`picture_convert`] изображений необходимо задавать пиксельные координаты источника и приемника изображения: `{src_coords}` и `{dst_coords}`. Как источником, так и приемником изображения в зависимости от конкретной команды могут являться окно экрана, файл или двоичная переменная. Если координаты явно не заданы (нулевые), то они полагаются равными размерам текущего окна или растра изображения. Размеры растра заносятся при записи изображения в файл (ширина и высота исходного изображения). Считается, что левый верхний угол изображения имеет пиксельные координаты (0,0), а размеры полного изображения описываются пиксельными координатами ((0,0), (<ширина растра>-1, <высота растра>-1)). Таким образом, задавая соответствующие пиксельные координаты, можно вывести на экран или заменить в файле произвольный фрагмент растра.

Изображение может быть записано в файл [`picture_save3`] либо в палитре источника, либо в палитре приемника (например, при замене фрагмента изображения в файле), задаваемой при записи. Если палитра приемника `{pal}` не задана, то по умолчанию палитрой приемника является текущая палитра экрана.

Изображение может быть записано в двоичную переменную только в формате VSF (без палитры), обеспечивающем быструю запись и быстрый показ изображения, что эффективно используется при запоминании и восстановлении фрагментов экрана [`screen_frame`].

Изображение может быть выведено на экран [`picture_show3`] или помещено либо в палитре источника, либо в палитре приемника `{pal}`, задаваемой при выводе (по умолчанию палитрой приемника является текущая палитра экрана). При выводе изображения на экран в палитре приемника может происходить подбор индексов, т.е. на экран выдается цвет из палитры приемника наиболее близкий к тому, которому соответствует цвет в палитре источника.

Преобразование форматов графических файлов [`picture_convert`] содержит в себе элементы чтения изображения из файла `m` записи преобразованного изображения в файл. Преобразованный растр записывается в файл вместе с палитрой источника, либо с палитрой приемника `{pal}`, явно указываемой при задании преобразования.

Из палитры приемника `{pal}` могут реально использоваться не все цвета, а только их некоторая часть, задаваемая индексами при помощи параметров `{fl}`, `{ind_ex}` и `{ind_use}`. Параметр `{ind_ex}` (при выводе изображения игнорируется) определяет диапазон цветов из палитры приемника, которые не изменяются в палитре выходного файла, при этом остальные цвета берутся из палитры входного файла. Параметр `{ind_use}` определяет номера цветов из палитры приемника, которые будут использоваться при подборе палитры.

Палитра источника, хранящаяся в файле изображения, может быть скорректирована с помощью таблицы перекодировки цветов `{fl, xlat}`.

Есть ли растр записан видео режиме, содержащим больше цветов, чем видео режим приемника, то для получения изображения (достаточно близкого по цветовой гамме к исходному) применяется режим растрингования `{fl}`, в котором происходит подбор цвета для каждого пикселя по определенному алгоритму с учётом цвета соседних пикселей. Улучшить полученное изображение можно путем подбора палитры по нескольким фрагментам изображения [`palette_make`].

Существует возможность создать файл [picture_make] одноцветного изображения произвольного формата в заданной палитре {pal} (по умолчанию в текущий палитре экрана).

При выводе изображения на экран [picture_show3] существует три вида масштабирования. Первый вид — автоматическое масштабирование, когда масштаб выводимого изображения изменяется в соответствии с размерами приемника (окна экрана). Второй вид — пропорциональное масштабирование, при котором сохраняется отношение ширины к высоте изображения источника в пределах размеров окна экрана. Третий вид — увеличение или уменьшение изображения в целое {scale} число раз. Конкретный вид масштабирования задается флагом {fl}, определяющим режим преобразования.

При получении изображений существует три способа наложения цветов, при которых изображение источника накладывается на изображение приемника. Первый способ — методика “прозрачного цвета”, в которой один выбранный цвет из налагаемых цветов источника объявляется “прозрачным” (или все цвета, кроме выбранного) {transp_col, FL_USE_TRANSP, FL_TC_ACTIV}. Второй способ — методика “фоновый цвет”, в которой один выбранный цвет из существующих цветов приемника сохраняется как фон (или все цвета кроме выбранного) {bckg_col, FL_USE_BACKG, FL_BG_ACTIV}. Третий способ — использование битовой маски, синхронной с источником или с приемником, при задании которой выводятся точки, соответствующие только единичным или только нулевым битам маски:

- {scale_xlen, FL_SCR_SYNCMASK, FF_SSM_ENABLE или
- {mask, FL_DST_SYNCMASK, FF_DSM_ENABLE}.

В режиме “размножение растра” возможно получение копий изображения с нужными характеристиками: без начального отступа первой копии растра по оси X, без циклического сдвига по оси X (копии располагаются строго одна под другой), с начальным отступом первой копии растра по оси X, с циклическим сдвигом по оси X {scr_pix_width, scr_pix_height, mjx, mju, mjxshift, mjxskip, FL_MULT}.

Режим “поворот на малый угол” обеспечивает поворот растра на угол в пределах от -15 до +15 градусов, при этом вносимые искажения зависят от величины угла поворота (при угле 11 градусов искажения составляют около 2%) {mjx, mju, mjxshift, mjxskip, FL_RRSA}.

6. Параметры, определяющие работу команд с изображениями. Приведем параметры, которые являются общими для команд преобразования изображений:

fl params pal pal_ex ind_ex ind_use mask xlat,

где fl — восьмеричное число, определяющее режим преобразования:

bit0(001) — алгоритм выбора координат экрана (приёмника или источника):

0 — брать координаты экрана из заданных параметров если они не заданы или все равны нулю, то берётся весь экран;

1 — брать в качестве координат экрана текущее окно;

bit1(002) — разрешение или запрет масштабирования по заданным координатам приемника;

bit2(004) — увеличение или уменьшение изображения в целое число раз, заданное в поле scale;

bits3-5(070) — формат файла (VSF,CEL,BMP,PCX,GIF,TIF,JPG);

bits6-8(0700) — количество битов на пиксель;

bits9-11(07000) — количество плоскостей;

bit12(010000) — использование чистых цветов или растрирование при показе несуществующих в текущей палитре;

bit13(020000) — подбор индексов или фиксация индексов палитры;

bit14(040000) — установка палитры приемника или фиксация палитры приемника;

bit15(0100000) — выдача информации о файле;

bit16(0200000) — использование для показа диапазонов индексов, заданных переменной ind_use, или всех индексов;

bit17(0400000) — запись в новый файл или в существующий файл (только BMP, VSF);

bit18(01000000) — стандартный режим или эмуляция индексного режима приемника;

bit19(02000000) — использование цветов, заданных в pal_ex, при подборе палитры;

bit20(04000000) — выбор палитры;

bit21(010000000) — изменение цветов либо всех индексов, либо индексов, заданных в ind_ex;

bit22(020000000) — переменная xlat игнорируется или используется;

bit23(040000000) — обычный режим или сохранение при масштабировании отношения ширины к высоте картинке;

bit24(0100000000) — обычный режим или центрирование картинке

params — имя двоичной переменной, содержащей параметры пересылки; ее формат достаточно сложен, поэтому рекомендуется в начале определить ее структуру с помощью команды prog_define, а

затем использовать именованный доступ к ее полям (см. описание ниже);

pal — имя двоичной переменной, содержащей палитру или таблицу перекодировки;

pal_ex — часть палитры, подлежащие сохранению;

ind_ex — индексы палитры приёмника, цвета которых нельзя изменять;

ind_use — индексы палитры приемника которыми можно рисовать или которые можно использовать при подборе подходящих цветов;

mask — имя двоичной переменной, содержащей битовую маску;

xlat — таблица перекодировки цветов;

Формат двоичной переменной params определяется через структуру RASTR_BLT_PAR с помощью команды proc_define:

```
proc_define RBLT
struct TKA_4S { int x1 y1 x2 y2; }
struct RASTR_BLT_PAR { long flags; TKA_4S dst_coords; \
    int dst_vstep; int dst_rowlen; int dont_tuch1[3]; \
    int dst_mem_type; long bckg_col; TKA_4S src_coords; \
    int src_vstep; int src_rowlen; int dont_tuch2[3]; \
    int src_mem_type; long trnsp_col; \
    int mjx mju mjvxshift mjxskip; int scale_xlen; \
    long src_pix_width src_pix_height; int scale; \
    long colconvmethod; }
#End_proc
```

Поля структуры RASTR_BLT_PAR имеют следующие значение:

flags — восьмеричное число, отдельные биты которого определяют режим работы команды:

FL_XLAT — признак использования переменной pal;

FL_SCALE — режим “масштабирование растра”;

FL_MULT — режим “размножение растра”;

FL_RRSA — режим “поворот на малый угол”;

FL_SRCSYNCMASK — режим “источник-синхронная маска”;

FL_DSTSYNCMASK — режим “приемник-синхронная маска”;

FL_USE_TRANSP — режим использования методики “прозрачного цвета” источника;

FL_USE_BACKG — режим использования методики “фонового цвета” приемника;

FL_SSM_ENABLE — способ интерпретации битовой маски, синхронной с источником;

FL_DSM_ENABLE — способ интерпретации битовой маски, синхронной с приемником;

FL_TC_ACTIVE — атрибут методики “прозрачного цвета”;

FL_TC_DSIGN — атрибут методики “прозрачного цвета” (для преобразования растра в битовую маску);

FL_BG_ACTIVE — атрибут методики “фонового цвета”;

FL_BG_DSIGN — атрибут методики “фонового цвета” (для преобразования растра в битовую маску);

dst_coords — координаты $x1, y1, x2, y2$ объемлющего прямоугольника в системе координат приемника; при размножении растра — область размножения, при повороте растра — координаты “неповернутого” прямоугольника, в остальных случаях — область вывода;

dst_vstep — вертикальный шаг между строками приемника в точках; при $dst_vstep > 1$ осуществляется чересстрочный вывод, используемый в GIF файлах;

dst_rowlen — длина строки приемника в байтах;

dst_mem_type — формат задания цвета пикселя приемника (при выводе на экран определяется автоматически);

bckg_col — цвет для использования в методике “фонового цвета”;

src_coords — координаты $x1, y1, x2, y2$ выводимой области в системе координат источника; координаты $x1, y1$ определяют первую выводимую точку;

src_vstep — вертикальный шаг между строками источника в точках; при $src_vstep > 1$ осуществляется чересстрочный ввод, используемый в GIF файлах;

src_rowlen — длина строки источника в байтах;

src_mem_type — формат источника (при выводе с экрана определяется автоматически);

trnsp_col — цвет для использования в методике “прозрачного цвета”;

mjx mju mjvxshift mjxskip — параметры размножения/поворота;

scale_xlen — длина в байтах строки источника синхронной битовой маски, используемой при выводе;
 src_pix_width, src_pix_height — полные размеры единичного растра в точках источника; используются только в режиме “размножение растра”;
 scale — параметр масштабирования растра;
 colconvmethod — указатель на функцию языка Си, осуществляющую преобразование цветов каждого пикселя.

Пример задания параметров в командах изображений через структуру RASTR_BLT_PAR:

```

foren#1 0 02 dt_pict1.bmp
* инициализация переменной par_show, соответствующей размерам
* структуры (для корректной работы перед установкой нужных
* полей; эта переменная должна быть нулевой)
bvar_ini par_show 0 0 0 104b
* задание через структуры координат приемника и источника
bvar_set par_show|(RASTR_BLT_PAR).dst_coords 0 4d 0 0 320 239
bvar_set par_show|(RASTR_BLT_PAR).src_coords 0 4d 0 0 100 100
* рисование "прозрачным цветом" 4
* отдельные битовые разряды флага задаются
* в глобальных переменных
= flg &(L &@FL_USE_TRANSP. &@FL_TC_ACTIVE. | &).
bvar_set par_show|(RASTR_BLT_PAR).flags 0 10 &flg.
bvar_set par_show|(RASTR_BLT_PAR).trnsp_col 0 1D 4
picture_show3 1120 #1 par_show

```

7. Команды для работы с изображениями. В Главном управляющем языке реализованы следующие команды для работы с изображениями:

palette_get3 — получение требуемой палитры (текущей, стандартной, серой, черной, белой и одноцветной);

palette_set3 — установка требуемой палитры;

palette_make — подбор палитры по нескольким фрагментам изображений;

picture_make — создание одноцветных файлов изображений произвольного формата (CEL, BMP, PCX, GIF, TIF, JPG, VSF);

picture_save3 — сохранение фрагмента экрана в файле в различных форматах (CEL, BMP, PCX, GIF, TIF, JPG, VSF) и в переменной формата VSF;

picture_save — сохранение фрагмента экрана в файле только в формате PCX;

picture_show3 — показ фрагмента на экране из файла одного из форматов (CEL, BMP, PCX, GIF, TIF, JPG, VSF) и из переменной формата VSF;

picture_show — показ фрагмента на экране из файла формата PCX;

screen_frame — быстрое сохранение фрагмента экрана в файле и переменной в формате VSF на непродолжительное время и восстановление его в пределах текущего видеорежима;

picture_convert — преобразование файла изображения из одного формата в другой (CEL, BMP, PCX, GIF, TIF, JPG, VSF) без участия экрана;

rastr_blt — быстрое преобразование изображения, находящегося в переменной в формате VSF или на экране (без использования файлов);

Авторы статьи участвовали в подготовке документации и тестировании команд для работы с изображениями. Разработку и реализацию этих команд осуществили Богомолов Н.А., Ковалёв А.Д., Краснов А.Н., Тимофеев Б.И.

Работа выполнена при поддержке РФФИ, проект № 98-07-90018.

СПИСОК ЛИТЕРАТУРЫ

1. Кенцл Т. Форматы файлов в Internet. С-П.: Питер, 1997.
2. Борн Г. Форматы данных. Киев: Торгово-издательское бюро BVH, 1995.
3. Арушанян О.Б., Богомолов Н.А., Волченкова Н.И., Ковалев А.Д., Синицын М.Н. Общее описание Главного управляющего языка Базовой технологии создания систем обработки и отображения сложно организованной информации // Библиотеки и пакеты прикладных программ. М.: Изд-во Моск. ун-та, 1996.
4. Фролов А.В., Фролов Г.В. Программирование видеоадаптеров. М.: Диалог-МИФИ, 1995.
5. Мюррей Дж.Д., Райпер У. Энциклопедия форматов графических файлов. Киев: BVH, 1997.

Поступила в редакцию
23.01.2000