

Михаил Краснов

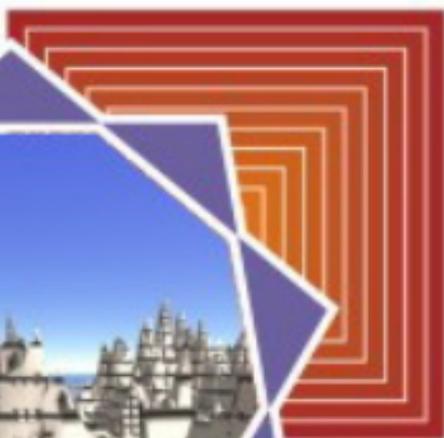


[www.bhv.ru](http://www.bhv.ru)  
[www.bhv.kiev.ua](http://www.bhv.kiev.ua)

# DirectX

## Графика в проектах *Delphi*

- Библиотеки DirectDraw и Direct3D
- Программирование игр
- Двумерная и трехмерная графика



# МАСТЕР

ПРАКТИЧЕСКОЕ РУКОВОДСТВО



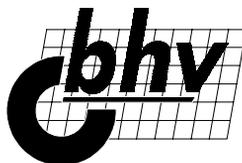
CD-ROM содержит примеры проектов, дистрибутив кдра DirectX 8.0a и демо-версию 3D Exploration

Михаил Краснов

# DirectX.

Графика в проектах

***Delphi***



*Санкт-Петербург*

Дюссельдорф ♦ Киев ♦ Москва ♦ Санкт-Петербург

УДК 681.3.06

Книга посвящена использованию модулей DirectX в приложениях, разрабатываемых в Delphi. Начиная с простых примеров, последовательно и подробно рассматривается создание объектов двумерной и трехмерной графики, визуальные и цветовые эффекты, а также обсуждаются дополнительные темы, такие как быстрая работа с устройствами ввода. Большое внимание уделяется вопросам оптимизации и ускорения работы приложений. Книга содержит практические решения проблем, возникающих при программировании игр и других приложений, нуждающихся в высокой скорости вывода графики в среде Windows. Прилагается компакт-диск с инструментальными средствами, кодами и демонстрационными версиями рассматриваемых примеров.

*Для широкого круга программистов*

**Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зав. редакцией	<i>Наталья Таркова</i>
Редактор	<i>Анна Кузьмина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн обложки	<i>Игоря Цырульникова</i>
Зав. производством	<i>Николай Тверских</i>

**Краснов М. В.**

DirectX. Графика в проектах Delphi. — СПб.: БХВ-Петербург, 2001. — 416 с.: ил.

ISBN 5-94157-033-3

© М. В. Краснов, 2001

© Оформление, издательство "БХВ-Петербург", 2001

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 06.06.01.

Формат 70×100<sup>1</sup>/<sub>16</sub>. Печать офсетная. Усл. печ. л. 33,54.

Тираж 4000 экз. Заказ

"БХВ-Петербург", 198005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар, № 77.99.1.953.П.950.3.99 от 01.03.1999 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов  
в Академической типографии "Наука" РАН.  
199034, Санкт-Петербург, 9-я линия, 12.

# Содержание

<b>Введение</b> .....	<b>7</b>
<b>Глава 1. Понятие о COM</b> .....	<b>11</b>
Библиотеки динамической компоновки .....	11
COM-модель .....	18
Контроль версии .....	19
COM-объекты .....	22
Интерфейсы .....	22
Что вы узнали в этой главе .....	30
<b>Глава 2. Обзор библиотеки DirectDraw</b> .....	<b>33</b>
Поверхности .....	33
Блиттинг .....	42
Буферы .....	53
Отладка приложений .....	58
Что вы узнали в этой главе .....	59
<b>Глава 3. Приемы использования DirectDraw</b> .....	<b>61</b>
Цветовой ключ .....	61
Полноэкранные приложения .....	66
Частичное обновление экрана .....	74
Непосредственный доступ к пикселям поверхности .....	76
Согласование содержимого буферов .....	81
Поворот изображения .....	83
Визуальные эффекты .....	87
Сохранение растровых изображений .....	92
Доступ к пикселям в 16-битном режиме .....	93
Полупрозрачность .....	98
Выбор объектов .....	104
Лупа .....	107
Палитры .....	110
Оконные приложения .....	113
Комбинированные приложения .....	118

Осциллограф .....	125
Что вы узнали в этой главе .....	128
<b>Глава 4. Спрайтовая анимация.....</b>	<b>129</b>
Спрайты.....	129
Хранитель экрана .....	136
Проверка столкновений.....	148
Спрайты и оконный режим .....	158
Что вы узнали в этой главе .....	159
<b>Глава 5. Пишем игру .....</b>	<b>161</b>
Оригинальный сплэш .....	161
Космический истребитель.....	164
Игра "Меткий стрелок" .....	168
Работа с клавиатурой .....	178
Работа с мышью .....	185
Вывод текста .....	189
Создание консоли .....	193
Диалоговые окна .....	196
Использование отсечения в полноэкранном приложении.....	200
Библиотека CDX .....	203
Что вы узнали в этой главе .....	211
<b>Глава 6. Работа с AVI-файлами.....</b>	<b>213</b>
Модуль <i>VFW</i> .....	213
Модуль <i>DirectShow</i> .....	217
Запись в видеофайл.....	219
Что вы узнали в этой главе .....	221
<b>Глава 7. Обзор библиотеки Direct3D.....</b>	<b>223</b>
Модуль <i>DirectXGraphics</i> .....	223
Тип <i>TColor</i> и цвет в Direct3D.....	231
Примитивы .....	232
Точки .....	236
Режимы воспроизведения .....	240
Блоки установок.....	242
Окрашенные вершины.....	243
Отрезки.....	248
Треугольник .....	252
Полноэкранный режим .....	262
Что вы узнали в этой главе .....	265
<b>Глава 8. Подробнее о библиотеке Direct3D.....</b>	<b>267</b>
Частичная прозрачность.....	267
Альфа-составляющая цвета .....	272
Размытие при движении.....	276
Работа с переменным числом вершин.....	278

---

Текстура.....	282
Текстурные координаты .....	288
Альфа-составляющая текстур .....	294
Мультитекстурирование.....	296
Цветовой ключ текстур.....	300
Спрайты в Direct3D .....	303
Что вы узнали в этой главе .....	307
<b>Глава 9. Трехмерные построения .....</b>	<b>309</b>
Матричный подход.....	309
Реалистичные изображения .....	316
Буфер глубины.....	320
Подготовка моделей.....	329
Что вы узнали в этой главе .....	356
<b>Глава 10. Визуальные эффекты .....</b>	<b>357</b>
Источник света и свойства материала .....	357
Туман .....	371
Двусторонние поверхности .....	374
Соприкасающиеся поверхности .....	375
Частичная прозрачность объемных фигур.....	377
Наложение текстуры на трехмерные объекты.....	379
Механизм трехмерной игры.....	382
Что вы узнали в этой главе .....	402
<b>Заключение.....</b>	<b>403</b>
<b>Приложение 1. Глоссарий .....</b>	<b>405</b>
<b>Приложение 2. Описание компакт-диска и требования к компьютеру .....</b>	<b>411</b>
<b>Список литературы .....</b>	<b>412</b>
<b>Предметный указатель.....</b>	<b>413</b>

# Введение

Главной темой книги, которую вы держите в руках, является компьютерная графика, а именно использование в Delphi модулей DirectX, связанных с двумерной и трехмерной графикой.

DirectX — это набор драйверов, образующий интерфейс между программами в среде Windows и аппаратными средствами. Состоит он из набора компонентов, поддерживающих непосредственную работу с устройствами, и служит в качестве средства разработки быстродействующих мультимедийных приложений. Для программиста применение DirectX заключается в использовании набора низкоуровневых интерфейсов (API).

Развитие DirectX происходит непрерывно и корпорация Microsoft ежегодно выпускает новую или обновленную версию этого продукта. Очередная версия включает в себя возможности предыдущих, но некоторые предлагают подходы, кардинально отличающиеся от концепций ранних версий. Так, в восьмой версии не произошло обновления модуля, связываемого с двумерной графикой, и разработчикам предложено использовать объединенный подход к графике, в котором чистая двумерная графика является частным случаем трехмерной. В этой версии единый набор API обслуживает оба подраздела компьютерной графики.

В данной книге обсуждается API седьмой и восьмой версий DirectX. В начале в ней изложено применение модуля DirectDraw для создания приложений чистой двумерной графики. DirectDraw используется как набор интерфейсов седьмой версии DirectX. Во второй части книги рассматривается компонент DirectX Graphics, как набор интерфейсов восьмой версии.

Читателю не стоит относиться к материалу книги о DirectDraw, как к устаревшему анахронизму. Во-первых, при работе с этим модулем приложения двумерной графики гарантированно используют возможности видеокарт, поддерживающих только 2D-акселерацию, а пользователей, имеющих именно такие карты, в ближайшие годы будет оставаться еще очень много.

Во-вторых, в планах разработчиков корпорации Microsoft было заявлено о том, что в девятой версии DirectX будет возобновлена поддержка DirectDraw, как обновленных интерфейсов, и этот материал наверняка будет легко "приспособить" также к новой версии. И, в-третьих, вы получите здесь представление о базовых механизмах и приемах, лежащих в основе и трехмерной графики, при переходе к которой вы встретите уже знакомые вам принципы и подходы.

Теперь я должен сказать несколько важных вещей непосредственно о книге.

Прежде всего, хочу предупредить, что книга не охватывает целиком ни DirectX, ни даже модули, относящиеся напрямую к графике. Материал чрезвычайно обширен, чтобы охватить его одним изданием, поэтому я вам не могу обещать, что вы после знакомства с моей книгой будете уметь абсолютно все, но я обещаю, что вы научитесь очень многому.

Как мне кажется, большинство читателей купят книгу в расчете на то, чтобы с ее помощью научиться создавать игры. Вы найдете в ней примеры простых игр, которые не стоит рассматривать как профессиональные. Но, познакомившись с ними, вы сможете написать и более масштабные проекты. Думаю, что книга может оказаться полезной и тем, кто не собирается программировать игры, а нуждается в средстве, позволяющем максимально быстро построить, например, диаграмму или график.

Не утверждаю, что я открыл новый жанр, но должен предупредить вас, что эта книга может показаться вам своеобразной: главный упор в ней делается на практические примеры. Среди прочитанных мною изданий по программированию самыми полезными оказались те, которые содержат не пространственные рассуждения и сложные диаграммы, а те, где предлагаются готовые решения и масса примеров. Поэтому и здесь я постарался выдержать изложение в том же духе в надежде, что она принесет вам действительную пользу. В книге масса примеров, многие из которых не стоит рассматривать слишком бегло, иначе в какой-то момент вы можете потерять нить понимания. По мере неспешного ознакомления с примерами в каждом из них попробуйте что-то изменить или добавить, и тогда у вас появится ощущение полного понимания.

Эта книга и примеры к ней также очень сильно отличаются от обычных книг по программированию на Delphi. Например, здесь нет обзора компонентов, в большинстве примеров на форме располагается один компонент, а код модулей может вам поначалу показаться непривычно длинным и странным по синтаксису.

Одна из целей, которую я преследовал, состоит в том, чтобы книга читалась легко теми, кто впервые сталкивается с данной темой. Но я должен предупредить, что если вы программируете на Delphi меньше года, вам, возможно, будет очень тяжело изучать эту книгу. Вы должны хорошо знать Delphi, причем подразумевается не умение ориентироваться в палитре компонентов,

а наличие опыта в кодировании. Минимальный уровень, который вы должны иметь, чтобы приступить к чтению этой книги, таков: читатель должен свободно владеть навыками работы с невизуальными объектами, такими как объекты класса `TBitmap`. Если вы можете с помощью подобного объекта вывести на форме содержимое растрового файла, то, я надеюсь, сможете легко и быстро разобраться в материале книги.

Наверняка вы слышаны о DirectX, и его существование не стало для вас откровением, пришедшим в вашу жизнь с этой книгой. Вы знаете, что данное средство предназначено для создания мультимедийных приложений, работающих максимально быстро, и у вас, наверное, нет вопроса ко мне, почему я написал книгу об использовании DirectX. Но, скорее всего, мне необходимо упомянуть, почему для освещения этой темы мною выбрана среда программирования Delphi. Ведь если в ней и написаны масштабные игры профессионального уровня, то их очень немного. Программисты, знающие среду Delphi поверхностно, несправедливо считают, что с ее помощью можно создавать только СУБД. А между тем, это очень мощное средство, которое годится для решения достаточно широкого круга задач. Прочитав книгу, вы убедитесь в этом. Delphi — очень популярная среда программирования, о которой разработчики DirectX позаботились не в первую очередь: для программистов, использующих C++ или Visual Basic, имеется богатый источник информации по разработке программ, комплект документации и примеров, SDK; для программистов же, использующих Delphi, таких источников информации мало. Чтобы помочь именно этой огромной армии программистов и написана данная книга. Это не руководство для тех, кто использует C++, или не умеет программировать вообще, но хочет научиться писать игры. Это учебник для тех, кто хорошо знает Delphi, но пока не умеет использовать DirectX.

Поскольку в Delphi отсутствует стандартная поддержка DirectX, нам приходится выбирать среди решений, предложенных сторонними разработчиками, главным образом, энтузиастами. Среди таких решений есть и привычное для Delphi, в виде наборов компонентов, например WDirectX и DelphiX. Но я предлагаю другое решение: мы будем использовать набор заголовочных файлов проекта JEDI. Это перенесенные энтузиастами заголовочные файлы из состава DirectX SDK корпорации Microsoft, изначально написанные на C. Такой подход хоть и приводит к кажущемуся поначалу чрезмерно громоздкому коду, но облегчит вам жизнь, когда, например, вы захотите разобраться в коде игр, написанных профессионалами. Очень многое для вас в чужом коде станет знакомым и понятным.

Обновления комплекта заголовочных файлов, а также дополнительные примеры использования DirectX в Delphi вы можете найти по ссылке <http://www.delphi-jedi.org/DelphiGraphics/>.

Заголовочные файлы, которыми я пользовался в настоящей книге, взяты мною с этого сайта, сведения об авторах трансляции указаны в коде модулей.

Здесь же вы можете найти файлы справки из состава DirectX SDK. Обратите внимание, что в файлах справки восьмой версии отсутствует информация о функциях DirectDraw, поэтому вам необходимо найти соответствующие файлы седьмой версии. Пока вы читаете первую главу книги, постарайтесь скачать эти файлы по указанному адресу.

Также вам после прочтения книги очень пригодятся переложения на Delphi примеров из DirectX SDK. Их вы найдете по адресу <http://groups.yahoo.com/group/JEDI-DirectXExamples>.

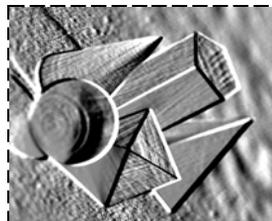
Прилагающийся к книге компакт-диск содержит инсталляцию ядра DirectX восьмой версии, но, возможно, к тому времени, когда вы начали читать эту книгу, уже появились новые версии. Их вы можете найти здесь:

- <http://www.microsoft.com/directx>
- <http://msdn.microsoft.com/directx>

По тем же адресам, наверняка, вы найдете и документацию по текущей версии.

Если у вас возникли какие-либо технические вопросы, такие, например, как проблемы с компакт-диск, обратитесь на сайт издательства <http://www.bhv.ru> (или [mail@bhv.ru](mailto:mail@bhv.ru)).

# ГЛАВА 1



## Понятие о СОМ

Глава является первой в серии глав, посвященных подсистеме DirectDraw. Эта библиотека, как и остальные модули DirectX, реализована в соответствии со спецификацией СОМ. Глава представляет собой краткий курс по этой спецификации и содержит минимальные сведения, необходимые читателю для понимания механизмов функционирования DirectDraw и других частей DirectX.

Примеры к данной главе располагаются в каталоге \Examples\Chapter01, для изучения они должны быть скопированы на диск, допускающий перезапись.

## Библиотеки динамической компоновки

Ключевым понятием операционной системы Windows, позволяющим понять любую технологию, используемую в ней, является понятие *библиотеки динамической компоновки* (DLL, Dynamic Link Library). Любое полноценное приложение этой операционной системы (32-разрядное приложение, имеющее собственное окно) использует DLL-файлы. По мере необходимости приложение обращается к библиотекам, вызывая из них нужные функции. Например, выполнимый модуль приложения не содержит кода по отображению окна, вывода в окно и реакции на большинство событий. Перечисленные действия реализуются в системных DLL. В частности, использованием такой технологии удастся экономить драгоценные ресурсы, один и тот же код не дублируется многократно, а размещается в памяти единожды.

К одной библиотеке, как правило, может обращаться одновременно несколько приложений. Библиотеку в такой схеме называют *сервером*, а обслуживаемое им приложение — *клиентом*. Сервером и клиентом в общем случае могут являться и библиотека, и приложение. В частности, это означает, что некоторая библиотека, в свою очередь, может "подгружать" функции из другой библиотеки.

Продемонстрируем работу операционной системы следующим примером. Создадим библиотеку, содержащую полезную функцию, выводящую на окне вызывающего клиента растровое изображение. Далее приведем инструкции ваших действий в среде Delphi. Готовый результат содержится в каталоге Ex01.

В главном меню выберите пункт **File | New** и в появившемся окне **New Items** щелкните на значке с подписью "DLL".

Чтобы выводимый растр не оказался легко доступным для посторонних глаз, скроем его, поместив в библиотеку. Для этого с помощью редактора ресурсов Image Editor (для вызова его выберите соответствующую команду меню **Tools**) создайте новый файл ресурсов с единственным ресурсом — нужным растром. Присвойте имя ресурсу — BMP1.

Для подготовки этого примера было взято одно из растровых изображений, поставляемых в составе пакета DirectX SDK, скопированное из окна редактора Microsoft Paint через буфер обмена.

Закончив редактировать растр, res-файл запишите в каталог, предназначенный для проекта библиотеки под именем DLLRes.res.

Код DLL-проекта приведите к следующему виду:

```
library Project1;    // Проект библиотеки
uses
  Windows, Graphics;
{$R DLLRes.res}    // Подключение файла ресурсов
// Описание экспортируемой функции, размещаемой в DLL (export) и
// вызываемой стандартно (stdcall)
procedure DrawBMP (Handle : THandle); export; stdcall;
var
  wrkBitmap : TBitmap;
  wrkCanvas : TCanvas;
begin
  wrkBitmap := TBitmap.Create;
  wrkCanvas := TCanvas.Create;
  try
    // Растр загружается из ресурсов, идентифицируется именем
    wrkBitmap.LoadFromResourceName (HInstance, 'BMP1');
    wrkCanvas.Handle := Handle;
    wrkCanvas.Draw(0, 0, wrkBitmap);
  finally
    wrkCanvas.Free;
    wrkBitmap.Free;
  end;
end;
```

```
// Список экспортируемых функций
// Функция у нас единственная
exports
    DrawBMP;
// Следующий блок соответствует инициализации библиотеки
begin
end.
```

Не будет лишним привести некоторые пояснения. Аргументом функции должен являться идентификатор канвы вызываемой формы. У вспомогательного объекта процедуры, класса `TCanvas`, значение этого идентификатора устанавливается в передаваемое значение, и теперь все его методы будут работать на канве окна, вызывающего функцию приложения.

А сейчас создайте DLL, откомпилировав проект.

### Внимание!

Откомпилируйте проект, но не запускайте его. Нельзя запустить DLL в понятии, привычном для обычного приложения.

В каталоге должен появиться файл `Project1.dll`. Исследуйте библиотеку: поставьте курсор на ее значок, нажмите правую кнопку мыши и в появившемся контекстном меню выберите команду **Быстрый просмотр**.

### Примечание

Если данная команда отсутствует, вам необходимо установить соответствующий компонент, входящий в дистрибутив операционной системы.

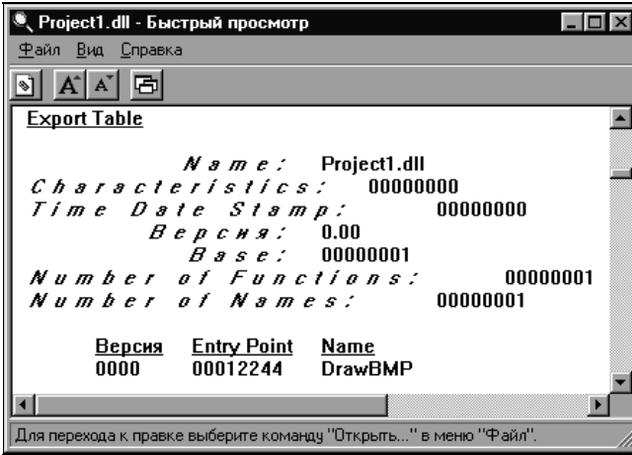
В окне отобразится информация о содержимом библиотеки, разбитая по секциям, среди которых нас особо интересует секция экспортируемых функций (рис. 1.1).

Если с помощью утилиты быстрого просмотра вы взглянете на содержимое модуля обычного приложения, то не найдете там секции экспортируемых функций. Это принципиальное отличие библиотек динамической компоновки от обычных исполняемых файлов.

### Примечание

Некоторые библиотеки скрывают секцию экспортируемых функций от обычного просмотра, но она там обязательно присутствует, даже если библиотека содержит только ресурсы. Пример подобной библиотеки — системная библиотека `moricons.dll`.

Итак, созданная нами библиотека содержит код экспортируемой функции с именем `DrawBMP` и растровое изображение. Сервер готов. Теперь создайте клиента. Организуйте новый проект, сохраните его в другом каталоге (готовый проект содержится в каталоге `Ex02`).



**Рис. 1.1.** Убеждаемся, что в секции экспортируемых функций созданной библиотеки присутствует название DrawBMP

В секции `implementation` введите следующую строку:

```
procedure DrawBMP (Handle : THandle); stdcall; external 'Project1.dll';
```

Этим мы декларируем нужную нам функцию. Ключевое слово `external` указывает, что данная функция размещена в библиотеке с указанным далее именем. Ключевое слово `stdcall` определяет вызов функции стандартным для операционной системы образом. При использовании импортируемых функций такие параметры задаются обязательно.

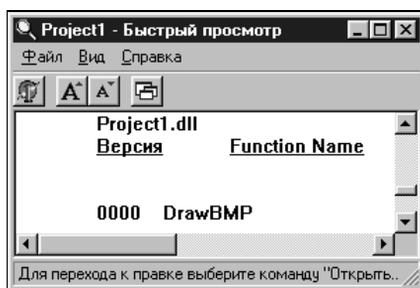
На форме разместите кнопку, в процедуре обработки события щелчка кнопки мыши которой введите строку:

```
DrawBMP (Canvas.Handle);
```

Аргументом вызываемой функции передаем ссылку канвы окна. Основной смысл этой величины — идентификация полотна окна.

Откомпилируйте проект, но пока не запускайте. С помощью утилиты быстрого просмотра исследуйте содержимое откомпилированного модуля: найдите в списке импортируемых функций следы того, что приложение использует функцию `DrawBMP` (рис. 1.2).

Обратите внимание, что имя известной нам функции находится в длинном ряду имен других импортируемых приложением функций. То есть модуль даже минимального приложения использует целый сонм функций, подключаемых из DLL. Именно о них упоминалось в начале главы как о функциях, ответственных за появление окна приложения и вывод на канве окна, а также отвечающих за реакцию окна на события. Эти функции именуются *системными*. Так же называются и библиотеки, хранящие код таких функций.



**Рис. 1.2.** Информация обо всех импортируемых приложением функциях доступна для анализа

### Примечание

Другое название системных функций — функции API (Application Program Interface).

При исследовании содержимого созданной нами библиотеки вы могли обратить внимание, что она ко всему прочему импортирует массу функций из системных библиотек.

Delphi позволяет нам писать краткий и удобочитаемый код, но при компиляции этот код преобразуется к вызову массы системных функций, и подчас одна строка кода "расшифровывается" вызовом десятка функций API. Программирование на Delphi образно можно себе представить как общение с операционной системой посредством ловкого переводчика, способного нам одной фразой передать длинную тираду, перевести без потери смысла, но некоторые потери мы все-таки имеем. Прежде всего, мы расплачиваемся тем, что приложения, созданные в Delphi, как правило, имеют сравнительно большой размер. Другая потеря — скорость работы приложения. При использовании библиотеки VCL и концепции объектно-ориентированного программирования вообще, мы жертвуем скоростью работы приложения.

В тех случаях, когда скорость работы приложения чрезвычайно важна, как в случае с обработкой графики, выход может состоять в том, чтобы отказаться от применения "переводчика", писать код, основанный исключительно на использовании функций API. Но такие программы плохо понятны новичкам, требуют специальной подготовки, поэтому мы не будем злоупотреблять этим. Если вы испытаете необходимость подробного разговора о том, как создавать в Delphi приложения без вызова библиотеки классов VCL, то автор может посоветовать вам свою предыдущую книгу, в списке литературы она поставлена на первое место. В ней вы найдете достаточно примеров подобных проектов. Ну а в этой книге постараемся не приводить таких примеров.

Вернемся к нашему примеру. Если вы сейчас запустите приложение, то вас постигнет неудача: сразу после запуска появится системное окно с сообще-