

Использование компонента **Image** для работы с графикой в **Delphi**

Василюк Ольга Александровна

"Брестский государственный университет имени А.С. Пушкина"

Delphi поддерживает три типа файлов – битовые матрицы, пиктограммы и мета-файлы. Все три типа файлов хранят изображения, различие заключается лишь в способе их хранения внутри файлов и в средствах доступа к ним. Битовая матрица (файл с расширением **.bmp**) отображает цвет каждого пиксела в изображении. При этом информация хранится таким образом, что любой компьютер может отобразить картинку с разрешающей способностью и количеством цветов, соответствующими его конфигурации.

Пиктограммы (файлы с расширением **.ico**) – это маленькие битовые матрицы. Они повсеместно используются для обозначения значков приложений, в быстрых кнопках, в пунктах меню, в различных списках. Способ хранения изображений в пиктограммах схож с хранением информации в битовых матрицах, но имеются и различия. В частности, пиктограмму невозможно масштабировать, она сохраняет тот размер, в котором была создана.

Метафайлы (**Metafiles**) хранят не последовательность битов, из которых состоит изображение, а информацию о способе создания картинка. Они хранят последовательности команд рисования, которые и могут быть повторены при воссоздании изображения. Это делает такие файлы, как правило, более компактными, чем битовые матрицы.

Delphi может работать со следующими типами файлов:

Тип файла	Расширение
JPEG Image File	.jpg, .jpeg
Битовые матрицы (Bitmaps)	.bmp
Пиктограммы	.ico
Enhanced Metafiles	.emf
Metafiles	.wmf

Компонент **Image** предназначен для отображения на форме графических изображений. По умолчанию выводит на поверхность формы изображения в **.bmp**-формате. Для вывода изображений в **.jpg**-формате необходимо в директиве **uses** подключить модуль **JPEG**.

Компонент **Image** активно используется во многих приложениях, причем не только для отображения статических картинок, но и для создания различных анимационных эффектов. Располагается компонент на вкладке **Additional**.

В большинстве случаев содержимое изображения загружается из файла на этапе проектирования. Загруженное в процессе проектирования из файла в компонент изображение не просто отображается на форме, но и сохраняется в приложении. Это дает возможность поставлять приложение без отдельного графического файла.

Компонент **Image** имеет различные свойства, расширяющие его возможности.

Кроме стандартных свойств, характерных многим визуальным компонентам (**Name**, **Height**, **Width**, **Align** и проч.), **Image** обладает специфическими свойствами:

Свойство	Характеристика свойства
AutoSize	Если установить свойство AutoSize в true , то размер компонента Image будет автоматически подгоняться под размер помещенной в него картинке. Если же свойство AutoSize установлено в false , то изображение может не поместиться в компонент или, наоборот, площадь компонента может оказаться много больше площади изображения.
Stretch	Позволяет подгонять не компонент под размер рисунка, а рисунок под размер компонента. Рисунок займет всю площадь компонента, но поскольку вряд ли реально вручную установить размеры Image точно пропорциональными размеру рисунка, то изображение исказится. Свойство Stretch не действует на изображения пиктограмм, которые не могут изменять своих размеров.
Center	Установленное в true центрирует изображение на площади Image , если размер компонента больше размера рисунка.
Picture	Определяет файл, из которого будет вставлен рисунок в компонент Image . Свойство Picture содержит отображаемую графическую составляющую, тип которой – битовая матрица, пиктограмма, метафайл или определенный пользователем тип.
Transparent	Если Transparent равно true , то изображение в Image становится прозрачным. Это можно использовать для наложения изображений друг на друга: компоненты Image размещаются так, чтобы они перекрывали друг друга, и в верхнем компоненте Transparent устанавливается равным true . Верхняя картинка перестанет заслонять нижнюю. При этом прозрачным (т.е. заменяемым на цвет расположенного под ним изображения) делается по умолчанию цвет левого нижнего пикселя битовой матрицы. Одно из возможных применений этого свойства – наложение на картинку надписей, выполненных в виде битовой матрицы. Такие надписи можно сделать с помощью встроенной в Delphi программы Image Editor .
Proportional	Позволяет автоматически масштабировать картинки без искажения. Для выполнения масштабирования значение свойства AutoSize должно быть равным False .

Загрузить изображение в компонент **Image** можно как статически, так и динамически. Статическая загрузка изображения осуществляется с помощью **Object Inspector**. Для этого необходимо нажать на три точки, расположенные справа от наименования свойства **Picture** компонента **Image**. Открывается окно **Picture Editor**, в котором после нажатия на кнопку «Загрузить (**Load**)» в раскрывшемся окне выбирается файл изображения.

Динамическая загрузка изображения в компонент **Image** осуществляется программно, например, при обработке некоего события. Строка кода

image1.picture.loadFromFile(имя файла)

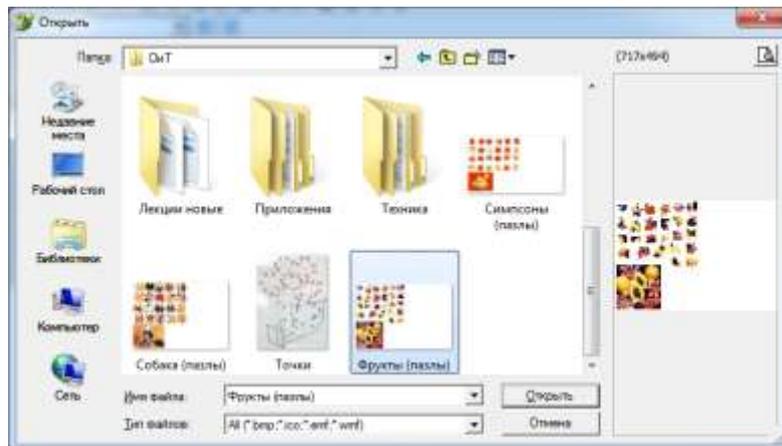
с помощью метода файлового чтения **LoadFromFile** осуществляет обращение к файлу в компоненте **Image1** по его имени.

Кроме метода файлового чтения **LoadFromFile**, объект **Image** обладает методом файловой записи **SaveToFile**.

Чтобы дать возможность пользователю просматривать и загружать любые графические файлы, необходимо:

1. Перенести на форму компонент **OpenPictureDialog** , расположенный в библиотеке на странице **Dialogs** и вызывающий диалоговое окно открытия и предварительного просмотра изображения:

2.



3. Выставить на форму кнопку, запускающую просмотр, или меню с единственным разделом **Файл**.

4. Написать оператор в обработчике щелчка на кнопке или на разделе меню:
if OpenPictureDialog1.Execute then
Image1.Picture.LoadFromFile(OpenPictureDialog1.FileName).

Этот оператор загружает в свойство Picture компонента **Image1** файл, выбранный в диалоге пользователем.