

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ  
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ»

## **ПРОГРАММИРОВАНИЕ В СРЕДЕ DELPHI**

Учебно-методическое пособие для вузов

Составители:  
И.Е. Воронина,  
Н.В. Огаркова

Издательско-полиграфический центр  
Воронежского государственного университета  
2011

## Оглавление

Введение.....	4
Теоретический материал .....	6
Задание 1 .....	21
Задание 2 .....	26
Задание 3 .....	29
Задание 4 .....	32
Задание 5 .....	40
Список литературы .....	41

## Теоретический материал

### Общая характеристика визуальных компонентов

Базовым для всех визуальных компонентов является класс `TControl`. Он обеспечивает основные функциональные атрибуты, такие как положение и размеры элемента, его заголовок, цвет и пр. В целом все визуальные компоненты можно разделить на две группы: оконные и неоконные элементы управления.

Оконный элемент управления представляет собой окно, предназначенное для решения конкретной задачи (кнопки, текстовые поля, полосы прокрутки). Базовым классом для оконных элементов управления является `TWinControl`. Оконные элементы управления содержат дескриптор окна (`window handler`). Дескриптором окна в операционной системе Windows называется 32-битная величина, однозначно определяющая данное окно. Приложение использует этот дескриптор для обращения к окну.

Для неоконных элементов управления базовым классом является `TGraphicControl`. Такие (неоконные) элементы управления не могут получать фокус ввода и быть родителями других компонентов. Достоинством неоконных элементов является то, что на них тратится меньше ресурсов (так как не нужен дескриптор окна). К таким компонентам относится, например, компонент `Label`.

### Свойства

Свойства позволяют управлять внешним видом и поведением компонентов при проектировании и выполнении приложения. Свойства компонентов, доступные при проектировании приложения, также доступны и при его выполнении. Существуют свойства, которые доступны только во время работы приложения.

`Name` – (тип `TComponentName`) – каждый компонент, помещаемый на форму, получает имя по умолчанию (`Label1`, `Edit1`). Это имя затем может быть использовано программистом для управления компонентом во время выполнения программы. Чтобы упростить процесс написания программы и сделать ее более читабельной, на начальном этапе разработки приложения изменить имя на более осмысленное, соответствующее назначению компонента. Примеры имен компонентов приведены в таблице 1:

Таблица 1

*Возможные значения свойства `Name`*

<code>Form1</code>	<code>fmMain</code>	<code>FormMain</code>
<code>Edit1</code>	<code>edName</code>	<code>EditName</code>
<code>Button1</code>	<code>btnOk</code>	<code>ButtonOk</code>

**Action** – (тип `TBasicAction`) – используется для синхронизации работы нескольких взаимосвязанных элементов управления (см. `TActionList`).

**Align** – (тип `TAlign`) – способ выравнивания компонента внутри контейнера, например, внутри формы (`TForm`) или панели (`TPanel`). Значения, которые может принимать данное свойство, приведены ниже:

- `alNone` – Так же, как и при проектировании приложения
- `alTop` – По верхнему краю
- `alBottom` – По нижнему краю
- `alLeft` – По левому краю
- `alRight` – По правому краю
- `alClient` – По клиентской области
- `alCustom` – Зависит от родительского компонента

**Caption** – (тип `TCaption`) – заголовок компонента. Один из символов заголовка может быть определен как «горячая» клавиша. Для этого перед символом необходимо указать знак «&». Например, если

`ButtonOk.Caption:=' &Ok'`

то нажатие клавиш `<Alt>+<O>` будет эквивалентно нажатию на кнопку `ButtonOk`.

**Color** – (тип `TColor`) – цвет фона (поверхности) компонента.

**Constraints** – (тип `TSizeConstraints`) – ограничения на размер компонента. Это класс со своими подсвойствами, наиболее важными из которых являются: `MinHeight`, `MaxHeight`, `MinWidth`, `MaxWidth`.

**Ctl3D** – (тип `boolean`) – задает вид визуального компонента (`true` – объемный, `false` – плоский).

**Cursor** – (тип `TCursor`) – вид указателя мыши (например, `crDefault`, `crNone`, `crCross` и т. д.).

**Enabled** – (тип `boolean`) – определяет доступность компонента, т. е. способность компонента реагировать на поступающие сообщения (`true` – доступен, `false` – недоступен). Блокировка относится только к пользователю, а программно этот компонент, его свойства и методы доступны.

**Font** – (тип `TFont`) – определяет шрифт текста. Основные свойства класса `TFont` приведены в таблице 2:

Таблица 2

*Основные свойства класса TFont*

Name	TFontName	Название шрифта (Arial, Times New Roman, ...)
Size	Integer	Размер шрифта в пунктах (1 пункт = 1/72 дюйма)
Height	Integer	Размер шрифта в пикселях (если значение положительное, то межстрочный интервал учитывается, если отрицательное – не учитывается)
Style	TFontStyle	Стиль шрифта. Свойство множественного типа, которое может иметь следующие значения: – fsItalic (курсив), – fsBold (полужирный), – fsUnderline (подчеркнутый), – fsStrikeOut (перечеркнутый)
Color	TColor	Цвет шрифта

Handle – (тип HWND) – дескриптор окна. Используется при вызове API функций Windows.

Height и Width – (тип Integer) – вертикальный и горизонтальный размеры компонента в пикселях.

Left и Top – (тип Integer) – координаты левого верхнего угла компонента относительно содержащего его контейнера.

HelpContext – (тип THelpContext) – номер раздела справочной системы. (0 – не задан.)

Hint – (тип String) – текст всплывающей подсказки. При этом свойство ShowHint должно быть выставлено в значение true.

PopupMenu – (тип TPopupMenu) – локальное всплывающее меню, связанное с компонентом. При этом свойство AutoPopup должно быть выставлено в значение true.

Text – (тип TCaption) – содержит строку, связанную с компонентом (содержимое компонента).

TabOrder – (тип TTabOrder) – порядок получения компонентами контейнера фокуса при нажатии клавиши <Tab>.

TabStop – (тип boolean) – возможность получения компонентом фокуса при нажатии клавиши <Tab>.