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

РАЗРАБОТКА КОМПОНЕНТОВ В DELPHI

Учебно-методическое пособие для вузов

Составители:
М.А. Артемов,
Г.Э. Вощинская,
В.Г. Рудалев

Издательско-полиграфический центр
Воронежского государственного университета
2011

Содержание

тупа к базам данных (TTable, TQuery и т.п.). Невизуальные компоненты не видны на форме и работают «за кулисами» приложения. Их предком не является TControl, а наследование идет от TComponent.

Визуальные компоненты делятся на две группы – графические и оконные. Графические компоненты порождаются от TGraphicControl и имеют визуальное представление, но не получают фокуса ввода. Главной особенностью потомков TGraphicControl является наличие свойства Canvas: TCanvas, позволяющего выводить изображение прямо на поверхности компонента. Примерами графических компонентов являются TImage, TPaintBox, TLabel.

Оконные компоненты происходят от TWinControl. Они инкапсулируют функции API для работы с окнами Windows, умеют получать фокус ввода и содержат свойство Handle: THandle. Данное свойство хранит дескриптор соответствующего оконного объекта Windows и может быть использовано для непосредственного вызова функций API и обмена сообщениями Windows между компонентами.

Иногда для создания пользовательского элемента управления используется класс TCustomControl. Это прямой потомок TWinControl, автоматически наследующий всю его функциональность. Но дополнительно он в секции Protected содержит свойство Canvas, необходимое для рисования. Чтобы использовать это свойство в потомках TCustomControl, его надо перенести в секцию Public (см. следующий раздел).

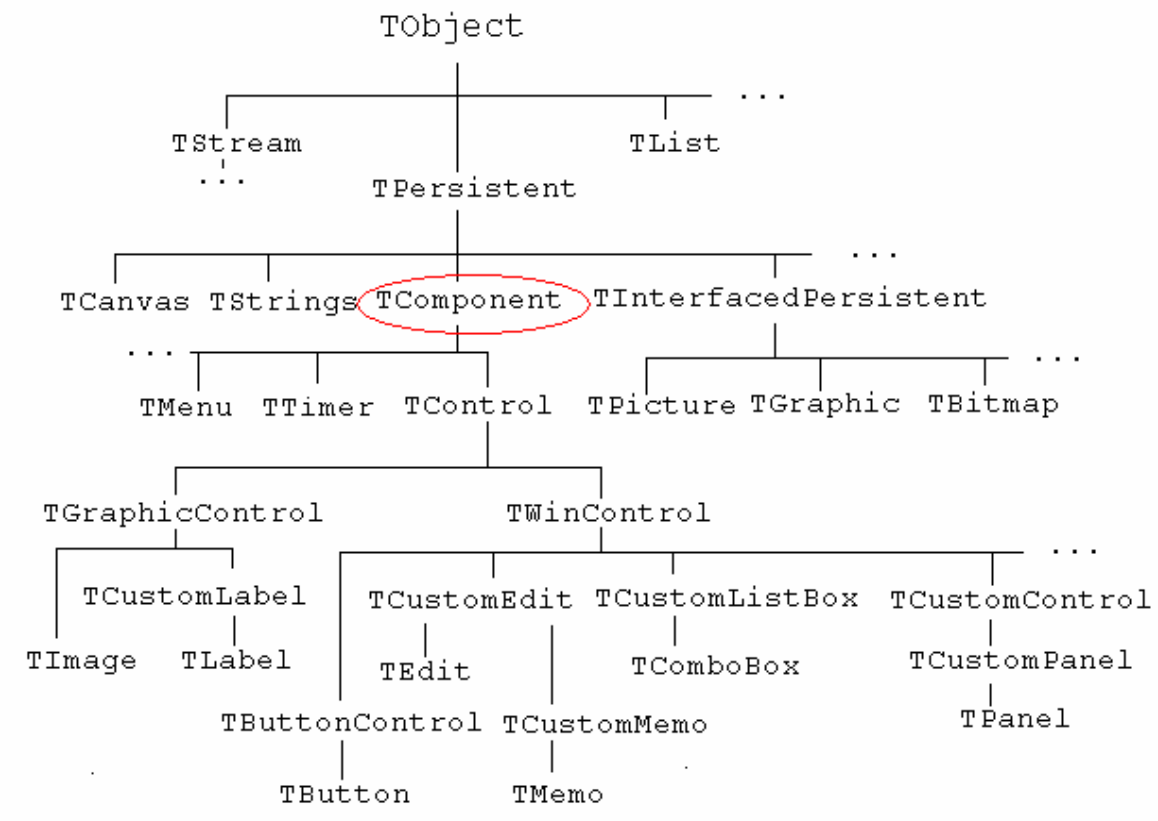

Рис. 1. Фрагмент иерархии классов библиотеки VCL

Рекомендации по использованию родительских классов в качестве предка для нового компонентного класса приведены в таблице 1.

Таблица 1

Базовые классы Delphi для создания компонентов

Унаследован от	Для создания
Существующего компонента	Модифицированной версии работающего компонента.
TGraphicControl	Графического компонента, не требующего фокуса ввода
TWinControl	Оконного компонента, использующего дескриптор окна или для создания оболочки вокруг существующих пользовательских оконных элементов управления.
TCustomControl	Оригинального компонента
TComponent	Невизуального компонента.

Зачастую новому компоненту не нужны все свойства предка, которых могут быть многие десятки. Чтобы упростить разработку, в иерархии компонентов имеются компоненты, в названии которых присутствует слово «Custom» (заказной). Такие компоненты не имеют опубликованных свойств, т.е. свойств, видных в инспекторе объектов. Все свойства, доступные к опубликованию, перечислены в секции `protected`. Поэтому если мы хотим модифицировать `TEdit`, но не все свойства `TEdit` нам нужны, то лучше наследовать не от `TEdit`, а от `TCustomEdit`. Далее, чтобы опубликовать необходимые свойства в нашем компоненте, следует при объявлении его класса просто перечислить названия свойств в секции `published`.

2. Структура компонента

2.1. Области видимости

Компоненты имеют разные программные интерфейсы: интерфейс времени выполнения (как компонент может использоваться, какие методы, свойства и события будут доступны), интерфейс времени проектирования (как компонент ведёт себя в среде проектирования), интерфейс разработчика (обеспечивает доступ к деталям реализации). Чтобы соответственно разграничить доступ к полям, методам и свойствам класса (то есть задать область видимости) в Delphi определены четыре основных директивы, приведенные в таблице 2.

Таблица 2

Уровни доступа к полям и методам класса

`Private`	Личные	Невидимы вне модуля или программы, в которой класс объявлен
`Protected`	Защищенные	Видим везде в модуле, в котором класс объявляется, и из любого класса потомка, независимо от модуля, где класс потомка появля-