

ФОРМА, КОМПОНЕНТЫ И ИХ СВОЙСТВА

План

1. Иерархия классов компонентов
2. Палитра компонентов Delphi 7
3. Основные элементы палитры компонентов
4. Вкладка Standart
5. Вкладка Additional
6. Вкладка Win32
7. Вкладка Dialogs
8. Вкладка Samples
9. Форма
10. Компоненты ввода и отображения текстовой информации. Кнопки, индикаторы, управляющие элементы.
11. Свойства компонентов

Ключевые понятия и термины: VCL, КОМПОНЕНТЫ, ПАЛИТРА КОМПОНЕНТОВ, LABEL, EDIT, BUTTON, RADIOBUTTON, ЧЕКБВОХ, ОБЪЕКТ ИНСПЕКТОР, ФОРМА.

Иерархия классов компонентов

Среда разработки **Delphi** ориентирована, прежде всего, на создание программ для Windows. При этом большое внимание уделяется возможности визуальной разработки приложений с помощью большого набора готовых **компонентов Delphi**, позволяющих избежать ручного кодирования. **Компоненты Delphi** охватывают практически все аспекты применения современных информационных технологий.

Прежде чем рассматривать основные компоненты Delphi, необходимо познакомиться с базовой иерархией классов этой системы.

1. Класс TObject – лежит в основе всей иерархии классов Delphi. Он обладает самыми общими методами, присущими любому объекту, описывает основные принципы поведения объектов во время работы программы (создание, уничтожение, обработка событий и др.).
2. Класс TPersistent – наследник класса TObject, реализует основные методы копирования содержимого объектов.
3. Класс TComponent – наследник класса TPersistent, основной родительский класс для всех классов, описывающих компоненты Delphi. В этот класс входит набор самых общих свойств, имеющих у каждого компонента, и некоторые полезные методы.
4. Класс TControl – наследник класса TComponent, служит основным классом для всех визуальных элементов управления. Если элемент управления является стандартным элементом Windows, то он базируется еще на одном промежуточном классе TWinControl (наследнике класса TControl).

Палитра компонентов

Палитра компонентов — это витрина той библиотеки компонентов — VCL или CLX, с которой вы работаете. При создании приложений для Windows вы работаете VCL. Но если вы создаете кросс-платформенное приложение, выполнив команду File | New | CLX Application, то палитра компонентов начинает отображать страницы CLX.

Палитра позволяет сгруппировать компоненты в соответствии с их смыслом и назначением. Эти группы или страницы снабжены закладками. Вы можете изменять комплектацию страниц, вводить новые страницы, переставлять их, вносить границы разработанные вами шаблоны и компоненты и т.д. Поскольку число predefined компонентов, конечно, возрастает от версии к версии, то наиболее полной является библиотека Delphi 7. Палитра этой библиотеки приведена на рис.1. По умолчанию в ней имеются страницы, перечисленные в табл. 1

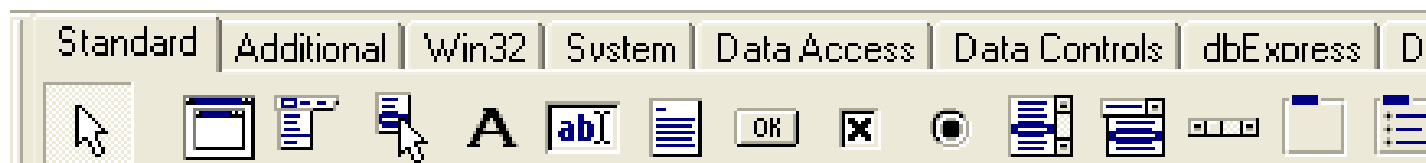


Рис. 1 Палитра компонентов

Таблица 1

Standard	Стандартная, содержащая наиболее часто используемые компоненты
Additional	Дополнительная, являющаяся дополнением стандартной
Win32	32-битные компоненты в стиле Windows 95/98/2000 и NT
System	Системная, содержащая такие компоненты, как таймеры, плееры и ряд других
Data Access	Доступ к данным
Data Controls	Компоненты отображения и редактирования данных
dbExpress	Связь с данными с помощью dbExpress(начиная с Delphi 6)
DataSnap	Компоненты для связи с сервером приложений при построении многопоточных приложений, работающих с данными (начиная с Delphi 6 и не во всех вариантах Delphi)

BDE	Доступ к данным через Borland Database Engine — BDE (начиная с Delphi 6, ранее эти компоненты размещались на странице Data Access)
ADO	Связь с базами данных через Active Data Objects (ADO) — множество компонентов ActiveX, использующих для доступа к информации баз данных Microsoft OLE DB (начиная с Delphi 5)
Interbase	Прямая связь с InterBase, минуя Borland Database Engine (BDE) и Active Data Objects (ADO) (начиная с Delphi 5)
WebServices	Компоненты клиентский приложений Web, использующие доступ к службам Web с помощью SOAP (начиная с Delphi 6)
InternetExpress	Построение приложений InternetExpress — одновременно приложений сервера Web и клиента баз данных с параллельными потоками (начиная с Delphi 5)
Internet	Компоненты для создания серверов Web
WebSnap	Компоненты для создания серверов Web, содержащих сложные страницы, управляемые данными (начиная с Delphi 6 и не во всех вариантах Delphi)
FastNet	Различные протоколы доступа к Интернет (только в Delphi 5 и 6; в Delphi 7 отсутствует)
Decision Cube	Компоненты для многомерного анализа данных (не во всех вариантах Delphi)
QReport	Компоненты для подготовки отчетов (начиная с Delphi 7 отсутствует)
Dialogs	Dialogs Диалоги, системные диалоги типа "Открыть файл» и др.
Win 3.1	Windows 3.x, компоненты в стиле Windows 3.x (оставлены для обратной совместимости с Delphi 1)
Samples	Образцы, различные интересные, но не до конца документированные компоненты
ActiveX	Примеры компонентов ActiveX
Rave	Компоненты для подготовки отчетов (начиная с Delphi 7)
IndyClients	Компоненты клиентских приложений Internet Direct (Indy), дающих доступ к различным протоколам Интернет из приложений Delphi, C++Builder, Kylix (начиная с Delphi 6)
IndyServers	Компоненты серверных приложений Internet Direct (Indy) (начиная с Delphi 6)
Indy Intercepts	Вспомогательные компоненты приложений Internet Direct (Indy) (начиная с Delphi 7)

Indy I/O Handlers	Вспомогательные компоненты приложений Internet Direct (Indy) (начиная с Delphi 7)
IndyMisk	Вспомогательные компоненты приложений Internet Direct (Indy) (начиная с Delphi 6)
COM+	Компонент, дающий доступ к каталогу COM+, содержащему данные по конфигурации COM+
InterBase Admin	Компоненты доступа к возможностям новой версии InterBase (начиная с Delphi 6)
IW Standard	Компоненты форм приложений IntraWeb (начиная с Delphi 5)
IW Data	Компоненты доступа к данным приложений IntraWeb (начиная с Delphi 7)
IW Client Side	Компоненты клиентских приложений IntraWeb (начиная с Delphi 7)
IW Control	Управляющие компоненты приложений IntraWeb (начиная с Delphi 7)
Servers	Оболочки VCL для распространённых серверов

Поскольку число страниц в Delphi 7 велико и не все закладки видны на экране одновременно, в правой части палитры компонентов имеются две кнопки со стрелками, направленными влево и вправо. Эти кнопки позволяют перемещать отображаемую на экране часть палитры. На некоторых страницах расположено столько компонентов, что они не помещаются в видимой части страницы. В этих случаях на концах страницы появляются дополнительные кнопки, позволяющие перемещаться вдоль страницы.

Чтобы перенести компонент на форму, надо открыть соответствующую страницу библиотеки и указать курсором мыши необходимый компонент. При этом кнопка-указатель, размещенная в левой части палитры компонентов, приобретёт вид не нажатой кнопки. Это значит, что вы находитесь в состоянии, когда собираетесь поместить компонент на форму. Поместить выбранный компонент на форму очень просто — надо сделать щелчок мышью в нужном месте формы.

Есть и другой способ поместить компонент на форму — достаточно сделать двойной щелчок на пиктограмме компонента, и он автоматически разместится на вашей форме. Если вы выбрали компонент, а затем изменили ваше намерение размещать его, вам достаточно нажать кнопку указателя. Это прервет процесс размещения компонента, и программа вернется в нормальный режим, в котором вы можете выбирать другой компонент или выполнять какую-то команду.

Имена компонентов, соответствующих той или иной пиктограмме, вы можете узнать из ярлычка, появляющегося, если вы задержите над этой пиктограммой курсор мыши. Если вы выберете в палитре компонент и нажмете клавишу F1, то вам будет показана справка по типу данного компонента. Тут надо сразу сделать замечание. Имена на ярлычках выглядят, например, так: **MainMenu**, **Button**. Однако в Delphi все имена классов в действительности начинаются с символа

«Т», например, **TMainMenu**, **TButton**. Под такими именами вы можете найти описания соответствующих компонентов во встроенной в Delphi справочной системе.

Вкладка Standart

Изучение Delphi естественным образом начинается со страницы палитры компонентов *Standart*. На этой странице расположены стандартные для *Windows* интерфейсные элементы, такие как главное и всплывающее меню, кнопка, однострочный и многострочный редакторы, переключатели, метки, списки, и некоторые другие компоненты, применяющиеся наиболее часто.



	Пикто- грамма	Имя	Назначение
1.	 фрейм	Frame	Панель с возможностями наследования. Проектируется как отдельное окно. Компонент визуальный.
2.	 главное меню	MainMenu	Позволяет конструировать и создавать полосу главного меню формы и выпадающие меню. Компонент невидимый.
3.	 всплыва ющее меню	PopupMenu	Позволяет конструировать и создавать всплывающие контекстные меню, возникающие при нажатии пользователем правой кнопки мыши. Компонент невидимый.
4.	 метка	Label	Используется для размещения на формах и других контейнерах текста, который не изменяется пользователем. Компонент визуальный.
5.	 окно редактир ования	Edit	Используется для ввода пользователем однострочных текстов. Может использоваться для отображения текста. Компонент визуальный.
6.	 многостр очное окно редактир ования	Memo	Используется для ввода и отображения многострочных текстов. Компонент визуальный.
7.	 командна я кнопка	Button	Используется для создания кнопок, которыми пользователь выбирает команды в приложении. Компонент визуальный.
8.	 контрольн ый индикатор с флажком	Checkbox	Позволяет пользователю включать и выключать различные опции. Компонент визуальный.
9.	 радиокно	RadioButton	Предлагают пользователю набор альтернатив, из которых выбирается одна. Набор реализуется

	пка		требуемым количеством радиокнопок, размещенных в одном контейнере (форме, панели и т.п.). Компонент визуальный.
10.	 окно списка	ListBox	Представляет собой стандартное окно списка Windows, позволяющее пользователю выбирать разделы из списка. Компонент визуальный.
11.	 редактируемый описок	ComboBox	Объединяет функции ListBox и Edit. Пользователь может либо ввести текст, либо выбрать его из списка. Компонент визуальный.
12.	 линейка прокрутки	Scrollbar	Представляет собой стандартную линейку прокрутки Windows и служит для управления положением видимой части форм или компонентов. Компонент визуальный.
13.	 групповое окно	GroupBox	Является контейнером, объединяющим группу связанных органов управления, таких как радиокнопки RadioButton, контрольные индикаторы CheckBox и т.д. Компонент визуальный.
14.	 группа радиокнопок	RadioGroup	Является комбинацией группового окна GroupBox с набором радио кнопок RadioButton; служит специально для создания групп радиокнопок. Можно размещать в компоненте несколько радиокнопок, но никакие другие органы управления не разрешены. Компонент визуальный.
15.	 панель	Panel	Является контейнером для группирования органов управления и меньших контейнеров. Панель можно использовать также для построения полос состояния, инструментальных панелей, палитр инструментов. Компонент визуальный.
16.	 панель	Panel	Является контейнером для группирования органов управления и меньших контейнеров. Панель можно использовать также для построения полос состояния, инструментальных панелей, палитр инструментов. Компонент визуальный.
17.	 список событий	ActionList	Обеспечивает диспетчеризацию событий компонентов. Компонент невизуальный

Вкладка Additional


На страницу **Additional** помещены дополнительные компоненты, без некоторых из которых сегодня трудно представить программу для Windows: кнопки с дополнительными свойствами, таблицы, компоненты для размещения изображений и многие другие, представленные в таблице:

	Пикто- грамма	Имя	Назначение
1.	 кнопка с графикой	BitBtn	Используется для создания кнопок, на которых располагается битовая графика (например, кнопка ОК с галочкой). Компонент визуальный.
2.	 кнопка с фиксацие й	SpeedButto n	Используется для создания инструментальных панелей и в других случаях, когда требуется кнопка с фиксацией нажатого состояния. Компонент визуальный.
3.	 маскиров анный ввод	MaskEdit	Используется для форматирования данных или для ввода символов в соответствии с шаблоном. Компонент визуальный.
4.	 таблица строк	StringGrid	Используется для отображения текстовой информации в таблице из строк и столбцов. Компонент визуальный.
5.	 таблица рисунков	DrawGrid	Используется для отображения в строках и столбцах нетекстовых данных. Компонент визуальный.
6.	 изображе ние	Image	Используется для отображения графики: пиктограмм, битовых матриц и метафайлов. Компонент визуальный.
7.	 формы	Shape	Используется для рисования фигур: квадратов, кругов и т.п. Компонент визуальный.
8.	 рамка	Bevel	Используется для рисования выступающих или утопленных линий или прямоугольных рамок. Компонент визуальный.
9.	 окно с прокрутк ой	ScrollBar	Используется для создания зон отображения с прокруткой. Компонент визуальный.
10.	 список с флажкам и	CheckListB ox	Компонент является комбинацией свойств списка ListBox и индикаторов CheckBox в одном компоненте. Компонент визуальный.
11.	 разделит ель панелей	Splitter	Используется для создания в приложении панелей с изменяемыми пользователем размерами. Компонент визуальный.
12.	 метка с бордюро м	StaticText	Компонент подобен компоненту Label, но обеспечивает дополнительные возможности по заданию стиля бордюра. Компонент визуальный.
13.	 инструме	ControlBar	Используется для размещения компонентов инструментальной панели. Компонент визуальный.

	Начальная панель		
14.	 события приложения	Application Events	Перехватывает события на уровне приложения. Компонент не визуальный.
15.	 диаграммы и графики	Chart	Компонент принадлежит к семейству компонентов TChart, которые используются для создания диаграмм и графиков. Компонент визуальный.

Вкладка Win32

Эта страница содержит компоненты, представляющие собой интерфейсные элементы для 32-разрядных операционных систем *Windows 95/98/NT* (В версии системы **Delphi 2** эта страница называлась *Win95*). Используя эти компоненты программы выглядят в стилистике последних версий операционных систем **Windows**.



	Пиктограмма	Имя	Назначение
1.	 страница с закладкой	TabControl	Позволяет организовывать страницы с закладками в стиле Windows 95, которые может выбирать пользователь. Компонент визуальный.
2.	 многопанельное окно	PageControl	Позволяет создавать страницы в стиле Windows 95/98, управляемые закладками или иными органами управления, для экономии места на рабочем столе. Компонент визуальный.
3.	 список изображений	ImageList	Предназначен для работы со списками изображений одинакового размера в меню, инструментальных панелях и т.п. Компонент не визуальный.
4.	 окно редактирования в формате RTF	RichEdit	Представляет собой окно редактирования в стиле Windows 95/98, позволяющее производить выбор цвета и шрифта, поиск текста и многое другое. Компонент визуальный.
5.	 ползунок	TrackBar	Управляющий элемент в виде ползунка в стиле Windows 95/98. Компонент визуальный.
6.	 отображение хода процесса	ProgressBar	Используется для отображения в стиле Windows 95/98 хода процессов, занимающих заметное время. Компонент визуальный.
7.	 кнопка-счетчик	UpDown	Кнопка-счетчик в стиле Windows 95/98 для ввода целых чисел. Компонент визуальный.

8.	 «горячие» клавиши	HotKey	Дает возможность реализовать в приложении поддержку горячих клавиш. Компонент визуальный.
9.	 воспроизведе ние немых клипов	Animate	Используется для воспроизведения немых клипов AVI, подобных используемым в Windows 95/98 изображениям копирования файлов и т.п. Компонент визуальный.
10.	 ввод дат и времени	DateTimePicker	Ввод дат и времени с выпадающим календарем. Компонент визуальный.
11.	 ввод дат	MonthCalendar	Ввод дат с выбором из календаря. Компонент визуальный.
12.	 дерево	TreeView	Предоставляет возможность просмотра структуры иерархических данных в стиле Windows 95/98. Компонент визуальный.
13.	 списки	ListView	Отображает списки в стиле Windows 95/98. Компонент визуальный.
14.	 заголовок	HeaderControl	Позволяет создавать составные перемещаемые заголовки в стиле Windows 95/98. Компонент визуальный.
15.	 полоса состояния	StatusBar	Полоса состояния программы, при необходимости — на нескольких панелях. Компонент визуальный.
16.	 инструмент альная панель	ToolBar	Инструментальная панель для быстрого доступа к часто используемым функциям приложения. Компонент визуальный.
17.	 инструмент альная перестраив аемая панель	CoolBar	Контейнер инструментальной панели, размеры которой могут изменяться пользователем. Компонент визуальный.
18.	 прокрутка страниц	PageScroller	Обеспечивает прокрутку больших окон, например, инструментальных панелей. Компонент визуальный.

Страница System


Страница System содержит компоненты, позволяющие использовать системные средства Windows.

	Пикто- грамма	Имя	Назначение
--	------------------	-----	------------

1.	 таймер	Timer	Используется для запуска процедур, функций и событий в указанные интервалы времени. Компонент невидимый.
2.	 окно для рисования	PaintBox	Используется для создания на форме некоторой области, в которой можно рисовать. Компонент визуальный.
3.	 аудио и видео плеер	MediaPlayer	Используется для создания панели управления воспроизведением звуковых и видео файлов, а также устройств мультимедиа. Компонент визуальный.
4.	 контейнер OLE	OLEContainer	Используется при создании области клиента для объекта OLE. Компонент визуальный.
5.	 диалог с сервером DDE	DDEClientControl	Используется клиентом DDE для организации диалога с сервером DDE. Компонент невидимый.
6.	 данные, передаваемые серверу DDE	DDEClientItem	Используется для определения данных клиента, передаваемых в диалоге серверу DDE. Компонент невидимый.
7.	 диалог с клиентом DDE	DDEServerControl	Компонент используется сервером DDE при проведении диалога с клиентом DDE. Компонент невидимый.
8.	 данные, передаваемые клиенту DDE	DDEServerItem	Компонент используется для определения данных сервера, передаваемых клиенту DDE в течение диалога. Компонент невидимый.

Вкладка Dialogs



Страница Dialogs содержит компоненты, используемые для создания различных диалоговых окон, общепринятых в приложениях Windows. Диалоги используются для указания файлов или выбора установок. Применение поставляемых в составе Delphi диалоговых окон помогает сэкономить время на разработку и придать вашему приложению совместимость с принятыми в Windows нормами диалога.






	Пиктограмма	Имя	Назначение
1.	 «Открыть файл»	OpenDialog	Предназначен для создания окна диалога «Открыть файл». Компонент невидимый.

2.	 «Сохранить файл как...»	SaveDialog	Предназначен для создания окна диалога «Сохранить файл как...». Компонент невидимый.
3.	 «Открыть рисунок»	OpenPicture Dialog	Предназначен для создания окна диалога «Открыть рисунок». Компонент невидимый.
4.	 «Сохранить рисунок как...»	SavePicture Dialog	Предназначен для создания окна диалога «Сохранить рисунок как...». Компонент невидимый.
5.	 «Шрифты»	FontDialog	Предназначен для создания окна диалога «Шрифты». Компонент невидимый.
6.	 «Цвет»	ColorDialog	Предназначен для создания окна диалога «Цвет». Компонент невидимый.
7.	 «Печать»	PrintDialog	Предназначен для создания окна диалога «Печать». Компонент невидимый.
8.	 «Установка принтера»	PrinterSetup Dialog	Предназначен для создания окна диалога «Установка принтера». Компонент невидимый.
9.	 «Найти»	FindDialog	Предназначен для создания окна диалога «Найти». Компонент невидимый.
10.	 «Заменить»	ReplaceDialog	Предназначен для создания окна диалога «Заменить». Компонент невидимый.

Вкладка Samples

Страница Samples содержит примеры компонентов. Поскольку это всего лишь примеры, они снабжены в Delphi лишь минимальной документацией и во встроенной справке сведения о них отсутствуют. Однако, исходные тексты примеров со страницы Samples имеются в каталоге ...\\Source\\Samples. Вы можете их просмотреть и понять, как построены эти примеры и как ими пользоваться.

	Пиктограмма	Имя	Назначение
1.	 индикатор хода процесса	Gauge	Пример компонента, используемого для создания индикатора хода процесса в виде линейки, текста или секторной диаграммы. Компонент визуальный.
2.	 таблица	ColorGrid	Пример компонента, используемого для создания таблицы цветов, в которой пользователь выбирает

	цветов		требуемый цвет. Компонент визуальный.
3.	 кнопка-счетчик	SpinButton	Пример компонента, используемого для создания кнопок-счетчиков. Компонент визуальный.
4.	 окно редактирования со счетчиком	SpinEdit	Пример компонента, используемого для создания окна редактирования в комбинации с кнопкой-счетчиком. Компонент визуальный.
5.	 дерево каталогов	Directory Outline	Пример компонента, используемого для отображения структуры каталогов выбранного диска. Компонент визуальный.
6.	 календарь	Calendar	Пример компонента, используемого для отображения календаря на указанный месяц в стандартном формате. Компонент визуальный.
7.	 индикатор события	IBEvent Alerter	Пример компонента, сигнализирующего о событии в базе данных. Компонент не визуальный.

Форма

Основой почти всех приложений Delphi является форма. Её можно понимать как типичное окно Windows. Форма является основой, на которой размещаются другие компоненты. Форма имеет те же свойства, что присущи другим окнам Windows. Она имеет управляющее меню в верхнем левом углу, полосу заголовка, занимающую верхнюю часть, кнопки разворачивания, свертывания и закрытия окна в верхнем правом углу. Можно изменить вид окна, убрав в кем какие-то кнопки или всю полосу заголовка, сделав его окном с неизменяемыми размерами и т.п. О том, как это сделать, вы узнаете с следующих уроках.

Во время проектирования форма покрыта сеткой из точек. В узлах этой сетки размещаются те компоненты, которые вы помещаете на форму. Во время выполнения приложения эта сетка, конечно, не видна.

В некоторых случаях при разработке какого-то модуля форма может оказаться вообще ненужной. Но обычно вся работа в Delphi проводится именно на форме.

Когда вы поместили на форме какие-то компоненты, вы можете получить по ним контекстную справку. Для этого выделите интересующий вас компонент и нажмите клавишу F1. Если щелкнуть на самой форме и нажать клавишу F1, будет показана справка по классу формы.

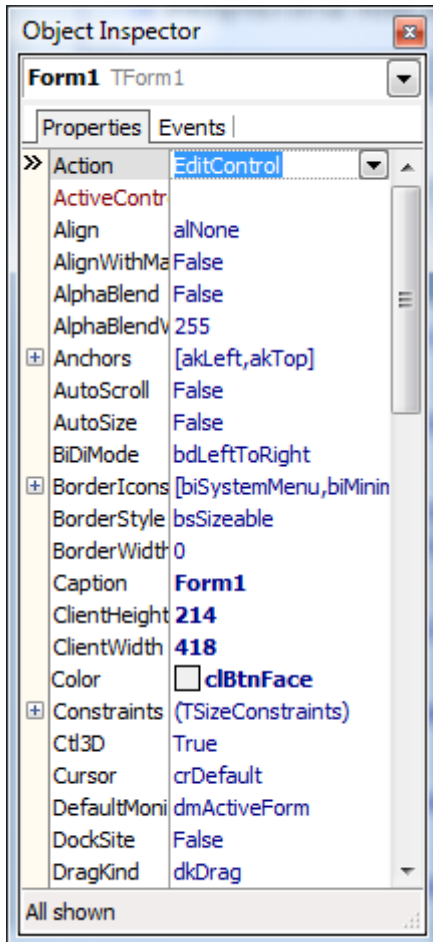
Компоненты ввода и отображения текстовой информации. Кнопки, индикаторы, управляющие элементы.

Для отображения текстовой информации в библиотеке визуальных компонентов Delphi существует множество компонентов.

1. Label (Метка) – надпись. Отображение текста, который не изменяется пользователем. Ее будем использовать для вывода комментариев на форму. Свойства: Caption (задается выводимый текст), Color, Top, Left и т.д.
2. Edit – строка ввода-вывода. Этот компонент можно использовать как для ввода данных – одного числа, одного символа, одной строки, так и для вывода этих же данных. Доступ ко всем данным, находящимся в окне, осуществляется с помощью свойства Edit1.Text. И для получения чисел необходимо использовать функции преобразование типов:
StrToInt(Edit1.Text) – преобразование строки в целое число;
StrToFloat(Edit1.Text) – преобразование строки в действительное число;
IntToStr(n) – преобразование целого числа в строку;
IntToFloat(x) – преобразование действительного числа в строку.
3. Memo – отображение, ввод и редактирование многострочных текстов. Основное свойство – Lines.
4. StringGrid – представляет собой таблицу, содержащую строки. Данные таблицы могут быть только для чтения или редактируемые. Для того, чтобы вывести с его помощью числовой массив, необходимо сначала преобразовать числовые данные в строки. Доступ к отдельной ячейке таблицы осуществляется с помощью следующего метода: StringGrid1.Cells[Col,Row]. Количество можно менять по необходимости. Для этого предназначены свойства StringGrid1.ColCount и StringGrid1.RowCount. Их значения можно менять как программно, так и в инспекторе объектов.
5. ListBox – отображение стандартного окна списка Windows, позволяющего из него выбирать пункты.
6. ComboBox – объединяет функции ListBox и Edit.
7. Button – кнопка. Используется для создания кнопок, которыми пользователь выполняет команды в приложении. Все действия, которые должны выполняться при нажатии на кнопку, записываются в процедуре обработки этого события.
8. RadioButton – радиокнопка – предлагает пользователю набор альтернатив, из которого выбирается одна.
9. RadioGroup – служит для создания групп радиокнопок.
10. CheckBox – флажок.

Свойства компонентов

Свойства компонента так же, как и свойства формы, можно изменить при помощи **Object Inspector**. Для того чтобы свойства требуемого компонента были выведены в окне **Object Inspector**, нужно выделить этот компонент (щелкнуть мышью на его изображении).



Action - определяет объект TAction . Это объект служит для быстрой привязки действий к компонентам, в особенности - к пунктам меню и панелям инструментов. Но может быть привязан и к форме. Для управления TAction служат редакторы TActionList со страницы Standard и TActionManager со страницы Additional .

ActiveControl - определяет элемент, который имеет в данный момент фокус ввода. Если выбрать какой-либо объект во время разработки (design-time), то при запуске приложения этот объект и будет иметь фокус ввода. Также свойство может быть полезно и во время выполнения (run-time) - можно узнать, какой объект "держит" фокус в данный момент, а также можно переместить фокус на любой из объектов. Пример: разместим на форме 2 кнопки - Button1 и Button2, а также TTimer (страница System). Выбрав элемент Timer1, дважды щёлкнем в Инспекторе объектов напротив надписи OnTimer на вкладке Events, т.е. создадим обработчик события и напишем следующее: ActiveControl:=Button2; Теперь, запустив программу, каждую секунду фокус будет перемещаться на Button2.

Align - определяет выравнивание формы на экране. Свойство принимает одно из следующих значений:

- alBottom - по нижнему краю;
- alClient - вся пользовательская (клиентская) область;

- **alCustom** - выравнивание определяется вызовом методом объекта-родителя;
- **alLeft** - по левому краю;
- **alNone** - без выравнивания;
- **alRight** - по правому краю;
- **alTop** - по верхнему краю.

AlphaBlend - включает/выключает прозрачность формы.

AlphaBlendValue - задаёт степень непрозрачности формы: 0 - форма полностью невидима, 255 - полностью видима. Прозрачность активируется только при установке свойства AlphaBlend в True.

anchors - определяет направления, по которым компоненты "привязываются" к форме. Пример: если установить у формы значения **akLeft** и **akRight** этого свойства в True, и точно также сделать у кнопки, то при изменении ширины формы размер кнопки (ширина) также будет изменяться.

AutoScroll - включает автоматическое появление полос прокрутки (Scroll bars) на форме, когда размеров формы недостаточно для отображения всех элементов.

AutoSize - включает автоматическое изменение размеров формы согласно позициям размещённых на ней элементов.

BiDiMode - определяет двунаправленное отображение элемента. В некоторых языках письмо осуществляется не слево-направо, а наоборот. Это свойство создано как раз для этой цели.

BorderIcons - определяет множество кнопок, которые отображаются в заголовке окна:

- **biSystemMenu** - единственный элемент, который не является кнопкой - отвечает за системное меню окна, которое вызывается комбинацией клавиш [Alt]+[Пробел].
- **biMinimize** - кнопка сворачивания (минимизации) окна;
- **biMaximize** - кнопка разворачивания окна;
- **biHelp** - кнопка справки.

Если хотя бы одна из кнопок сворачивания и разворачивания включена, то независимо от состояния другой, отображаются обе (но вторая естественно неактивна). Если выключены обе, они не отображаются вообще. Это не зависит от Delphi - так устроена ОС Windows.

BorderStyle - определяет поведение границ окна и общий тип окна:

- **bsDialog** - диалоговое окно (из кнопок - только "Закреть", иконки в заголовке окна нет);

- **bsNone** - "чистый лист" (отсутствие у окна границ и заголовка) - применяется обычно для создания заставок во время запуска программы;
- **bsSingle** - обычное окно, но с запретом изменения размеров;
- **bsSizeable** - обычное окно (по умолчанию) - размеры формы можно изменять;
- **bsSizeToolWin** - упрощённое окно с уменьшенным заголовком;
- **bsToolWindow** - упрощённое окно с уменьшенным заголовком без возможности изменения размеров.

BorderWidth - ширина границы окна в пикселах. Граница является невидимой и расположена в пользовательской части формы.

Caption - текст заголовка формы.

ClientHeight, ClientWidth - размер клиентской (пользовательской) части формы, т.е. той, на которой располагаются компоненты.

Color - цвет формы.

Constraints - определяет минимальные и максимальные размеры высоты и ширины формы в пикселах. 0 - любое значение, т.е. без ограничений.

Ctl3D - свойство определяет 3D-вид формы. При выключенном - "плоское" изображение.

Cursor - курсор мыши в тот момент, когда он находится над формой.

DefaultMonitor - определяет, на каком мониторе появится форма. Имеет смысл применять это свойство только при наличии более, чем одного монитора (например, если несколько экранов).

DockSite, DragKing и **DragMode** - определяют поведение формы при осуществлении операций Drag&Drop.

Enabled - отвечает за общую активность формы. Если установлено в False, форма недоступна.

Font - шрифт, используемый на форме.

FormStyle - стиль формы или её поведение в MDI-приложении (многооконное приложение, где дополнительные формы располагаются "внутри" основной формы). Значения:

- **fsNormal** - обычная форма (значение по умолчанию);
- **fsMDIChild** - дочерняя (подчинённая) форма MDI-приложения;
- **fsMDIForm** - главная форма MDI-приложения;
- **fsStayOnTop** - форма находится поверх всех окон на экране.

Height - высота формы в пикселах. В отличие от ClientWidth является высотой с учётом заголовка и границ формы.

HelpContext, HelpFile, HelpKeyword, HelpType - свойства для связи формы с файлом справки в формате *.hlp .

Hint - текст всплывающей подсказки.

HorzScrollBar - свойство определяет внешний вид и поведение горизонтальной полосы прокрутки окна.

Icon - значок (иконка) формы. Отображается в заголовке слева от заголовка. Задаётся файлом в формате *.ico .

KeyPreview - если свойство установлено в True, то при нажатии клавиш сначала будут вызываться обработчики формы, а только затем обработчики того компонента, который в данный момент имеет фокус ввода. События, связанные с нажатием клавиш - OnKeyDown(), OnKeyPress(), OnKeyUp().

Left - позиция формы на экране (левого верхнего угла) в пикселах.

Menu - позволяет выбрать один из компонентов-меню, который станет главным меню окна, т.е. будет отображаться вверху.

Name - имя формы как объекта. Может содержать только латинские буквы, цифры и знак подчёркивания, и не может начинаться с цифры. Фактически, это то имя, по которому в программе можно обратиться к форме.

ObjectMenuItem - используется при работе с OLE-объектами и позволяет связать пункт меню и OLE-объект: когда объект выделен, пункт меню активен и наоборот.

OldCreateOrder - определяет, когда происходят события OnCreate() и OnDestroy() формы. Если установлено в False, то OnCreate() произойдёт после вызова всех конструкторов, а OnDestroy() - после вызова всех деструкторов. Начальное значение - False, изменять не рекомендуется.

ParentBiDiMode - изменение свойства BiDiMode согласно значению объекта-предка формы.

ParentFont - изменение шрифта (Font) согласно значению объекта-предка.

PixelsPerInch - пропорции шрифта в системе (точек на дюйм).

PopupMenu - позволяет указать контекстное меню (объект TPopupMenu) для формы. Это меню вызывается нажатием правой кнопки мыши.

Position - определяет начальную позицию формы на экране, т.е. в момент её появления. Основные значения:

- **poDesigned** - появление в том месте, в каком форма расположена в design-time;
- **poDesktopCenter** - по центру рабочего стола (рекомендуемое значение);
- **poScreenCenter** - по центру экрана;
- **poMainFormCenter** - по центру главной формы приложения (для главной формы не имеет смысла).

PrintScale - определяет размеры формы при выводе её изображения на печать.

Scaled - включает масштабирование формы в соответствии с заданным значением свойства **PixelsPerInch**.

ScreenSnap - если установлено в **True**, то форма будет автоматически "прилипать" к краям экрана в момент перемещения.

SnapBuffer - определяет расстояние (в пикселах), на котором форма будет "прилипать" к краю экрана.

ShowHint - включает/выключает показ всплывающей подсказки (**Hint**).

Tag - специальное свойство, которое есть у всех объектов. Специального применения для этого свойства нет, поэтому оно используется для разных целей в конкретной ситуации. Свойство удобно в том случае, если нужно хранить некоторое целое число - не придётся заводить дополнительную переменную.

Top - позиция формы (левого верхнего угла) на экране в пикселах.

TransparentColor - включает/выключает прозрачность определённого цвета формы.

TransparentColorValue - задаёт цвет, который будет прозрачным.

UseDockManager - используется при реализации **Drag&Drop** технологии, предоставляя дополнительные возможности этого метода взаимодействия.

VertScrollBar - определяет внешний вид и поведение вертикальной полосы прокрутки окна.

Visible - определяет видимость формы на экране.

Width - ширина окна в пикселах, включая границы.

WindowMenu - свойство-аналог свойства **Menu**, но используемое при создании **MDI**-форм.

WindowState - одно из состояний окна:

- wsNormal - обычное состояние (занимает часть экрана);
- wsMinimized - окно свёрнуто;
- wsMaximized - окно развёрнуто на весь экран.

Вопросы и задания

1. Что из себя представляет и где находится палитра компонентов?
2. Назовите основные страницы палитры компонентов?
3. Каким образом можно переместить элемент на форму?
4. Назовите основные элементы находящиеся на вкладке Standart.
5. Какие компоненты располагаются на вкладке Additional?
6. Какие компоненты располагаются на вкладке Dialog?
7. Что такое "Форма"?
8. В каком окне редактируются свойства компонентов?
9. Какое свойство отвечает за видимость компонента на экране?
10. Что из себя представляет компонент StringGrid?
11. Назовите стили форм.
12. Каким образом можно сделать полупрозрачную форму?
13. Какой компонент служит для отображения статического текста?
14. Какой компонент используется для отображения многострочных текстов?
15. Как начинаются имена классов в Delphi?