22 Охарактеризуйте назначение системных диалогов

В приложениях часто приходится выполнять стандартные действия: откры­вать и сохранять файлы, задавать атрибуты шрифтов, выбирать цвета палитры, производить контекстный поиск и замену и т.п. В состав Delphi включены простые компоненты, реализующие соответствующие диалоговые окна Windows. Они размещены на странице **Dialogs**.

* OpenDialog - «Открыть файл»;
* SaveDialog - «Сохранить файл»;
* OpenPictureDialog - «Открыть рисунок»;
* SavePictureDialog - «Сохранить рисунок»;
* FontDialog - «Шрифты»;
* ColorDiolog - «Цвет»;
* ColorBox - выбор цвета;
* PrintDialog - «Печать»;
* PrinterSetupDialog - «Установка принтера».

Все компоненты стандартных диалогов со страницы Dialogs являются невизуальными компонентами, следовательно, их размещения на форме не имеет значения. При обращении к ним вызываются стандартные диалоги, вид кото­рых зависит от версии Windows и настройки системы. Так что при запуске одного и того же приложения на компьютерах с разными системами диалоги будут выгля­деть по-разному. Например, при русифицированной версии Windows все их надпи­си будут русскими, а при англоязычной версии надписи будут на английском язы­ке.

Основной метод, которым производится обращение к любому диалогу, — **Execute**. Эта функция открывает диалоговое окно и, если пользователь произвел в нем какой-то выбор, то функция возвращает **true**. При этом в свойствах компонента — диалога запоминается выбор пользователя, который можно прочитать и использо­вать в дальнейших операциях. Если же пользователь в диалоге нажал кнопку От­мена или клавишу **Esc**, то функция **Execute** возвращает **false**. Поэтому стандартное обращение к диалогу имеет вид:

if <имя компонента-диалога> Execute then <операторы,, использующие выбор пользователя>;

Компоненты **OpenDialog** — диалог «Открыть файл» и **SaveDialog** — диалог ♦Сохранить файл как», пожалуй, используются чаще всего, в большинстве прило­жений.

Все свойства этих компонентов одинаковы, но их смысл несколько различен для открытия и сохранения файлов. Основное свойство, в котором возвращается в виде строки выбранный пользователем файл, — **FileName**. Значение этого свойства можно задать и перед обращением к диалогу. Тогда оно появится в диалоге как значение по умолчанию в окне **Имя файла**

Типы искомых файлов, появляющиеся в диалоге в выпадающем списке Тип файла, задаются свойством **Filter**. В процессе проектирования это свойство проще всего задать с помощью редактора фильтров, который вызывается нажатием кнопки с многоточием около имени этого свойства в Инспекторе Объек­тов. При этом открывается окно редактора. В его левой панели **Filter Name** вы записываете тот текст, который увидит пользователь в выпадающем списке Тип файла диалога. А в правой панели **Filter** записываются разделенные точками с запятой шаблоны фильтра.

Свойство **Filterlndex** определяет номер фильтра, который будет по умолчанию показан пользователю в момент открытия диалога. Например, значение **Filterlndex = 1** задает по умолчанию первый фильтр.

Свойство **InitialDir** определяет начальный каталог, который будет открыт в момент начала работы пользователя с диалогом. Если значение этого свойства не задано, то открывается текущий каталог или тот, который был открыт при последнем обращении пользователя к соответствующему диалогу в процессе выполнения данного приложения.

Свойство **DefaultExt** определяет значение расширения файла по умолчанию. Если значение этого свойства не задано, пользователь должен указать в диалоге полное имя файла с расширением. Если же задать значение **DefaultExt**, то пользователь может писать в диалоге имя без расширения. В этом случае будет принято заданное расширение.

Свойство **Title** позволяет вам задать заголовок диалогового окна. Если это свойство не задано, окно открывается с заголовком, определенным в системе (например, «Открытие файла» в окне на рис. 3.56). Но вы можете задать и свой заголовок, подсказывающий пользователю ожидаемые действия. Например, «Укажите имя открываемого файла».

В компонентах диалогов открытия и сохранения файлов предусмотрена возможность обработки ряда событий. Такая обработка может потребоваться, чтобы установить все диктуемые конкретным приложением ограничения на выбор файлов. Событие **OnCanClose** возникает при нормальном закрытии пользователем диалогового окна после выбора файла. При отказе пользователя от диалога — нажатии кнопки Отмена, клавиши Esc и т.д. событие **OnCanClose** не наступает. В обработке события **OnCanClose** вы можете произвести дополнительные проверки выбранного пользователем файла.

Можно также написать обработчики событий **OnFolderChange** — изменение каталога, **OnSelectionChange** — изменение имени файла, **OnTypeChange** — изменение типа файла. В этих обработчиках вы можете предусмотреть какие-то сообщения пользователю.