Компоненты списка

ListBox

ComboBox

Компонент **Delphi ListBox** это компонент, служащий для хранения и обработки текстовой информации. Каждая строка **Delphi ListBox** хранит строку данных в формате **String**.

Компонент ListBox - это массив строк. **ListBox** может загружать данные с диска, и сохранять информацию в файл. Также ListBox может сортировать строки. Доступ к строкам предоставляет свойство**Items** этого компонента.

В соответствии с этим, вот как происходит обращение к строке с номером i:

ListBox1.Items[i-1]; // Минус один, так как нумерация строк в компоненте начинается с нуля.

Назад

Далее

Пользователь может выделить строку, щёлкнув по ней мышкой. Номер выделенной строки возвращает свойство компонента ListBox.ItemIndex. То есть, получить текст выделенной строки можно так:

S := ListBox1.Items[ListBox1.ItemIndex];

Для удаления строк из компонента применяется метод Delete. Удаление строки с номером i:

ListBox1.Items.Delete(i);

Чтобы при добавлении каждая строка была отсортирована (сортировка идёт только по возрастанию), нужно установить:

ListBox1.Sorted := True;

Также возможно установить это свойство на этапе конструирования в Инспекторе Объектов.

Компонент ListBox автоматически добавит полосу прокрутки, если количество строк не помещается по высоте компонента. Высота каждой строки равна **ItemHeight**. Если нужно чтобы при добавлении новой строки полоса прокрутки точно не возникла, этот размер нужно добавить к высоте компонента.

Назад

Далее

Методы добавления строк в компонент **Delphi ListBox**

Считывание из файла	**Компонент Delphi ListBox может обращаться напрямую к текстовому файлу как для считывания информации из файла, так и для сохранения всех своих строк в текстовый файл. Каждая запись в файле будет записана в виде одной строки компонента, и при сохранении каждая строка будет сохранена в виде одной записи файла:**
	ListBox1.Items.LoadFromFile('Имя_Файла'); // Процедура считывания из файла
	ListBox1.Items.SaveToFile('Имя_Файла'); // Процедура записи в файл

Назад

Далее

Добавление строки в конец списка	**Программист может последовательно добавлять строки в компонент, не заботясь об их нумерации, и они будут размещаться в конец списка:**
	ListBox1.Items.Add('Новая_Строка'); //Добавление строк в конец списка
	Так как количество строк в компоненте равно ListBox1.Items.Count, то новая (последняя) строка имеет номер ListBox1.Items.Count-1. Это объясняется тем, что нумерация строк начинается от 0.

Назад

Далее

Добавление строки перед строкой с номером i	**Программист может разместить новую строку среди существующих строк там, где ему необходимо. Для этого нужно воспользоваться методом Insert, и указать номер строки, перед которой необходимо записать новую строку:**
	ListBox1.Items.Insert(i, 'Новая_Строка'); // Добавление строки перед строкой с номером i
	При этом новая строка получает номер i.

Назад

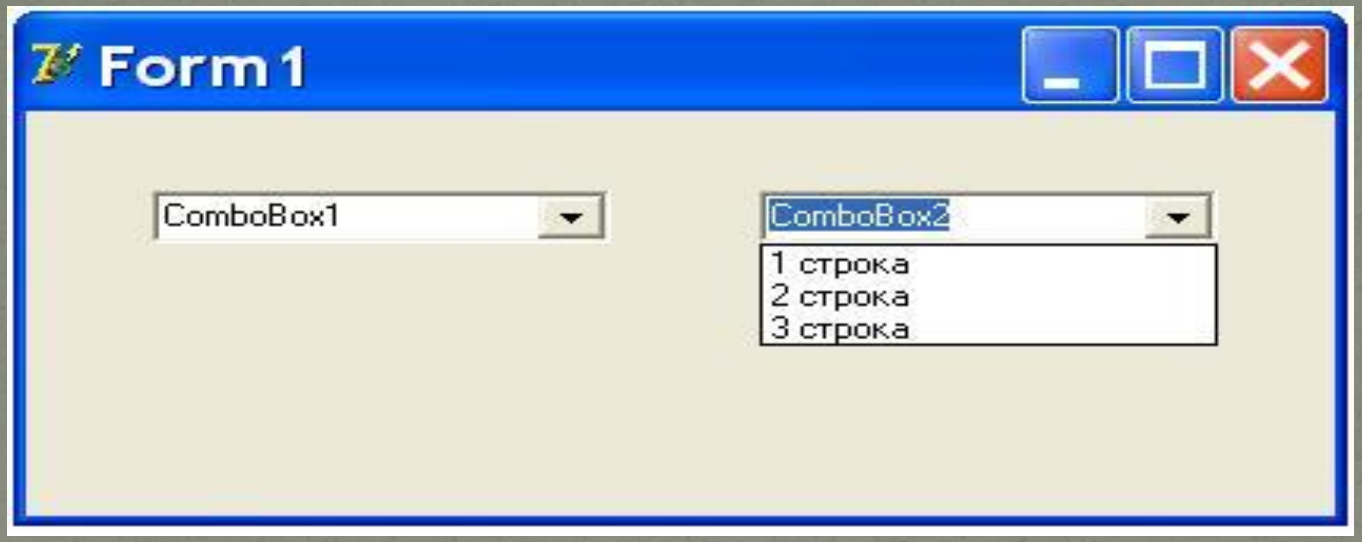
Далее

Добавление строк на этапе конструирования	**В Инспекторе Объектов зайдите на строчку Items и нажмите на появившуся кнопку. Появится окно редактора содержимого, где и можно внести нужную информацию. После нажатия кнопки OK содержимое редактора появится в компоненте ListBox на Форме.**

Назад

Вернуться в начало

Компонент Delphi ComboBox представляет собой комбинацию списка строк ListBox со строкой ввода Edit. При этом "список строк" компонента Delphi ComboBox вначале скрыт, и раскрывается при щелчке мышкой по треугольничку раскрытия, который находится справа в строке ввода:



Назад

Далее

Таким образом, с помощью **Delphi ComboBox** место на Форме экономится для расмещения других элементов интерфейса программы. А при необходимости раскрытие списка строк можно вообще запретить.

Многие свойства и возможности компонента **Delphi ComboBox** по работе со строками (объектами**Items**) такие же как и у компонента Delphi ListBox.

Отличия и дополнительные возможности.

Итак, за возможность раскрытия компонента ComboBox отвечает стиль - свойство **Style**:

Стиль компонента ComboBox1 равен
ComboBox1.Style = csDropDown
(список может быть раскрыт, можно писать в строке ввода)
Стиль компонента ComboBox2 равен
ComboBox1.Style = csSimple
(список не может быть раскрыт, можно писать в строке ввода)

Назад

Далее

При стиле **csDropDownList** компонент может быть раскрыт, но свойство **Text** может принимать значения только одной из строк, сохранённых в компоненте, то есть пользователь лишён возможности писать в строке ввода.

Количество строк, видимых при раскрытии списка, равно **DropDownCount**. Если реальное количество сохранённых строк больше этого количества, то автоматически появляется полоса прокрутки. Программист может заставить список раскрыться в нужный момент. Для этого нужно свойству**DroppedDown** присвоить значение **True**. Для этой же цели пользователь может использовать клавиатурную комбинацию **ALT+ВНИЗ.**

Максимально допустимое количество символов в текстовой строке задаётся параметром **MaxLength**, причём значение **0** означает отсутствие ограничений. Свойство **CharCase** управляет преобразованием вводимого текста к верхнему (значение CharCase = **ecUpperCase**) или нижнему (значениеCharCase = **ecLowerCase**) регистру. Значение **ecNormal** означает, что текст вводится без преобразования.

Назад

Далее

Основными событиями компонента **Delphi ComboBox** являются:

Событие	Условие генерации
OnChange	Изменился текст в строке ввода.
OnDropDown	Список раскрывается. Это событие нужно обработать, если содержимое списка может изменяться во время работы программы. Тогда в обработчике этого события можно заново сформировать содержимое списка.

Назад

Вернуться в начало