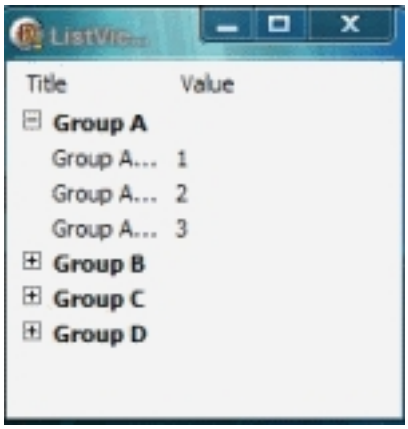
Если Вы используете Delphi 2007, Delphi 7 или предыдущую версию, Вам может понадобиться группировка в **TListView**.

Идея состоит в том, чтобы подражать раскрывающимся спискам со значками + и -.

Пункты в списке представлены объектом **TListItem**, которое определяет заголовок пункта, его изображение и другое визуальное поведение пункта.

Обратите внимание, что выполнение группировки, представленное здесь оставляет достаточно места для добавления Ваших собственных потребностей.



Чтобы выполнить это, поместите **ListView** на форму и установите **ViewStyle** = *vsReport*.

Для простоты, у нас будет два столбца (**"Title"** и **"Value"**).

Каждый элемент списка будет назначен на объект **TGroupItem** или **TItem**. Элемент **TG roup**
содержит группу пунктов
TItem
. Элементы
Group

будут развернуты или свернуты, чтобы показать или скрыть подпункты.

Нормальные пункты будут описаны в классе **TItem**:

```
TItem = class  private    fTitle: string;    fValue: string;  public    constructor Create(const title, value : string) ;    property Title: string read fTitle;    property Value : string read fValue;  end;
```

Элемент, который будет использоваться, чтобы содержать коллекцию пунктов, описан классом **TGroupItem**:

```
TGroupItem = class  private    fItems : TObjectList;    fCaption: string;    fListItem: TListItem;  fExpanded: boolean;    function GetItems: TObjectList;  public    constructor Create(const caption : string;                const numberOfSubItems : integer) ;    destructor Destroy; override;  procedure Expand;    procedure Collapse;      property Expanded : boolean read fExpanded;  property Caption : string read fCaption;    property Items : TObjectList read GetItems;    property ListItem : TListItem read fListItem write fListItem;  end;
```

FillListViewGroups добавляет пункты в список:

```
  procedure TForm1.FillListViewGroups;      procedure AddGroupItem(gi : TGroupItem) ;    var  li : TListItem;    begin      li := lvGroups.Items.Add;        li.Caption := gi.Caption;  li.ImageIndex := 1; //collapsed        li.Data := gi;      gi.ListItem := li; //link "back"    end;  begin  ClearListViewGroups;      AddGroupItem(TGroupItem.Create('Group A', 3)) ;  AddGroupItem(TGroupItem.Create('Group B', 1)) ;    AddGroupItem(TGroupItem.Create('Group C', 4)) ;    AddGroupItem(TGroupItem.Create('Group D', 5)) ;  end;
```

Пункт списка добавляется в список при помощи его свойства **Data**, указывающим на экземпляр **TGroupItem**

В конструкторе мы добавляем фиктивные подпункты:

```
constructor TGroupItem.Create(const caption: string;      const numberOfSubItems : integer) ;  var    cnt : integer;  begin    fCaption := caption;      for cnt := 1 to numberOfSubItems do    begin      Items.Add(TItem.Create(            caption + ' item ' + IntToStr(cnt), IntToStr(cnt))) ;  end;  end;
```

Обратите внимание, что каждый объект **TItem**, добавленный к свойству **TGroupItem** будет представлен как нормальный пункт. Затем, нам нужны функции, чтобы разворачивать и сворачивать группы.

Используется событие **OnDblClick** управления **TListView** для обработки этого:

```
//handles TListView OnDblClick event  procedure TForm1.lvGroupsDblClick(Sender:
TObject) ;  var    hts : THitTests;    gi : TGroupItem;  begin    hts := lvGroups.GetHitTestInfoAt(
      lvGroups.ScreenToClient(Mouse.CursorPos).X,
lvGroups.ScreenToClient(Mouse.CursorPos).y) ;     if (lvGroups.Selected  nil) then    begin     if
TObject(lvGroups.Selected.Data) is (TGroupItem) then     begin      gi :=
TGroupItem(lvGroups.Selected.Data) ;       if NOT gi.Expanded then       gi.Expand       else
      gi.Collapse;     end;    end;  end;
```

Дважды нажатый пункт определяется, используя метод **GetHitTestInfoAt** для вызова методов **Expand** или **Collapse**.

Метод **Expand** отобразит подпункты, а метод **Collapse** их сотрет.

Вот метод **Expand**:

```
procedure TGroupItem.Expand;  var    cnt : integer;    item : TItem;  begin    if Expanded
then Exit;     ListItem.ImageIndex := 0;    fExpanded := true;     for cnt := 0 to -1 + Items.Count
do   begin     item := TItem(Items[cnt]) ;     with TListView(ListItem.ListView).Items.Insert(1 +
cnt +          ListItem.Index) do     begin      Caption := item.Title;
SubItems.Add(item.Value) ;       Data := item;       ImageIndex := -1;     end;    end;  end;
```

Событие **OnAdvancedCustomDrawItem** используется, чтобы вывести полужирное изображение элементов группы.

TImageList используется, чтобы хранить изображения "+" и "-", которые указывают развернутое или свернутое состояние группы.

Исходник можно скачать [здесь.]