ЛЕКЦИЯ №5

# 1. ТЕХНОЛОГИЯ ПЕРЕТАСКИВАНИЯ ОБЪЕКТОВ «DRAG & DROP»

# Технология «Drag & Drop» («Перетащи и кинь») предназначается для перемещения (перетаскивания) данных из одних визуальных компонент в другие в процессе работы программы с помощью мыши. Данная технология делает интерфейс программы более удобным и быстрым и используется во многих системных и прикладных Windows-программах.

# Технология «Drag & Drop» позволяет, например, копировать строку текста (пункт) из одного компонента формы «ListBox» в другой компонент «ListBox» или предложение из одного компонента «Memo» в другой компонент «Memo». Технология «Drag & Drop» позволяет перемещать данные из любого визуального компонента в любой другой визуальный компонент для любой формы текущего приложения. С окнами других приложений данная технология по умолчанию не работает.

# Для перетаскивания данных из одних визуальных компонент в другие с помощью технологии «Drag & Drop» необходимо на этапе создания программы настроить компоненты на данную технологию и запрограммировать алгоритм их поведение при перетаскивании из (в) них данных.

# Для того, чтобы компонент смог «откликаться» (выделять перетаскиваемый компонент другим цветом и менять форму курсора) на начало процесса перетаскивания из него данного, необходимо его свойству «DragMode» присвоить значение «dmAutomatic». После этого компонент сможет предоставлять выделенное данное в распоряжение технологии «Drag & Drop» для перетаскивания.

Edit1.DragMode:=dmAutomatic; ListBox1.DragMode:=dmAutomatic;

# Компонент, давший данное для перетаскивания, отдает управление Widows и выходит из участия из процесса перетаскивания. Далее Windows следит за тем, куда пользователь перемещает данное по компонентам формы. Как только данное будет находится над каким-нибудь визуальным компонентом, Windows вызовет у него событие «OnDragOver» и передаст в соответствующую ему процедуру - обработчик данное о том, от кого (какого компонента) прибыло перетащенное данное. Компонент, над которым находится перетаскиваемое данное, в процедуре-обработчике события «OnDragOver» анализирует его и если ему оно нужно, то устанавливает параметру процедуры «Accept» значение «true», или значение «false» в противном случае. Значение «Accept» равное «true» озночает, что компонент не возражает принять «зависшее» над ним данное.

procedure TForm1.Edit1DragOver(Sender, Source: TObject; X, Y: Integer;

State: TDragState; var Accept: Boolean);

{ Можно принять (едиту), если "кидается" на него метка или данные из лист-бокса}

begin Accept := (Source is TEdit) or (Source is TListBox); end;

# Если данное переместится на другой компонент, или пользователь отменит перетаскивание клавишей клавиатуры «Esc», то компонент «забывает» о «зависшем» над ним данным и нечего больше не предпринимает. Если же пользователь отпустил данное над компонентом, то у компонента возникает событие «OnDragDrop».

# В обработчик события «OnDragDrop» передается данное о том, кто послал данное и в какое место компонента его кинули. Именно данная процедура обработчик (у компонента-приемника данного) реализует алгоритм перемещения данного с другого компонента в данный. Операционная система Windows в данной технологии служит только посредником для визуального отображения процесса перетаскивания, и не участвует в фактической обработке процесса переноса данных.

# В процедуре - обработчике события «OnDragDrop» необходимо программно реализовать алгоритм перемещения информации из одного визуального компонента, в другой.

procedure TForm1.Edit1DragDrop(Sender, Source: TObject; X, Y: Integer);

{ Обработка факта "кидка" для едита; Sender - на кого кинули; Source - что кинули (кого) }

begin { Если кинули из лист-бокса, то ...}

if Source is TListbox then with Sender as TEdit do begin

Text:=(Source as TListbox).items[(Source as TListbox).itemindex];

{ если кинули из другого едита , то … }

end else (Sender as TEdit).Text:=(Source as TEdit).text;

end;