

## Лабораторна робота № 18

Написання програм з використанням механізму перетягання об'єктів.

### Основні теоретичні відомості

#### *Механізм перетягання Drag-and-Drop.*

Для того, щоб працював механізм переносу Drag-and-Drop необхідно настроїти два елементи керування, один з яких повинен бути джерелом (Source) другий - приймачем (Target).

Любий елемент керування з палітри компонентів Delphi є джерелом в механізмі Drag-and-Drop. Його поведінка на початковому етапі переносу залежить від значення властивості

*type*

*TDragMode = (dmManual, dmAutomatic);*

*property*

*DragMode: TDragMode;*

Значення dmAutomatic забезпечує автоматичну реакцію компонента на натискання лівої кнопки миші та початок переносу.

Для ініціалізації переносу використовується метод:

*procedure BeginDrag(Immediate: Boolean; Threshold: Integer = -1);*

Параметр immediate = True забезпечує миттєвий старт механізму. При значенні False механізм включається лише при переміщенні курсору на відстань, що визначається параметром Threshold.

Про включення механізму свідчить вказівник миші - він змінюється на курсор, який визначається властивістю

*property DragCursor: TCursor;*

Джерело при переміщенні курсору не змінює власного положення, а тільки у випадку успішного завершення переносу може взаємодіяти з приймачем.

Приймачем може стати будь-який компонент, у якого створений метод-обробник

*procedure DragOver(Source: TObject; X, Y: Integer; State: TDragState; var Accept:*

*Boolean);*

Він викликається при переміщенні курсору у режимі Drag-and-Drop над цим компонентом.

Якщо параметр Accept отримує значення True, то даний компонент стає приймачем.

Джерело переносу визначається параметром Source. Через цей параметр розробник має доступ до властивостей та методів джерела. Біжуче положення курсору задають параметри X та Y. Параметр state повертає інформацію про характер руху миші:

*type*

*TDragState = (dsDragEnter, dsDragLeave, dsDragMove);*

dsDragEnter - вказівник з'явився над компонентом; dsDragLeave - вказівник покинув компонент; dsDragMove - вказівник переміщається по компоненту.

Приймач повинен передбачати виконання деяких дій, якщо джерело завершить перенос саме на ньому. Для цього використовується метод - обробник

*type*

*TDragDropEvent = procedure(Sender, Source: TObject; X, Y: Integer) of object;*  
*property*

*OnDragDrop: TDragDropEvent;*

який викликається при відпусканні лівої кнопки миші на компоненті - приймачі.

Доступ до джерела та приймача забезпечують параметри Source та Sender відповідно.

Координати миші повертають параметри X та Y. При завершенні переносу елемент керування

- джерело - отримує відповідне повідомлення, яке обробляється методом

*type*

*TEndDragEvent = procedure(Sender, Target: TObject; X, Y: Integer) of object;*  
*property*

*OnEndDrag: TEndDragEvent;*

Джерело та приймач визначаються параметрами Sender та Target відповідно. Для програмної зупинки

переносу можна використати метод EndDrag джерела: *procedure EndDrag(Drop: Boolean);*

Параметр Drop = True завершує перенос. Значення False перериває перенос.

Механізм перетягання здійснюється у декілька етапів:

1. Фіксується початок перетягання за допомогою метода форми BeginDrag (при натисканні лівої клавіші миші над об'єктом, вказівник миші приймає вид, що є вказаним у властивості DragCursor).

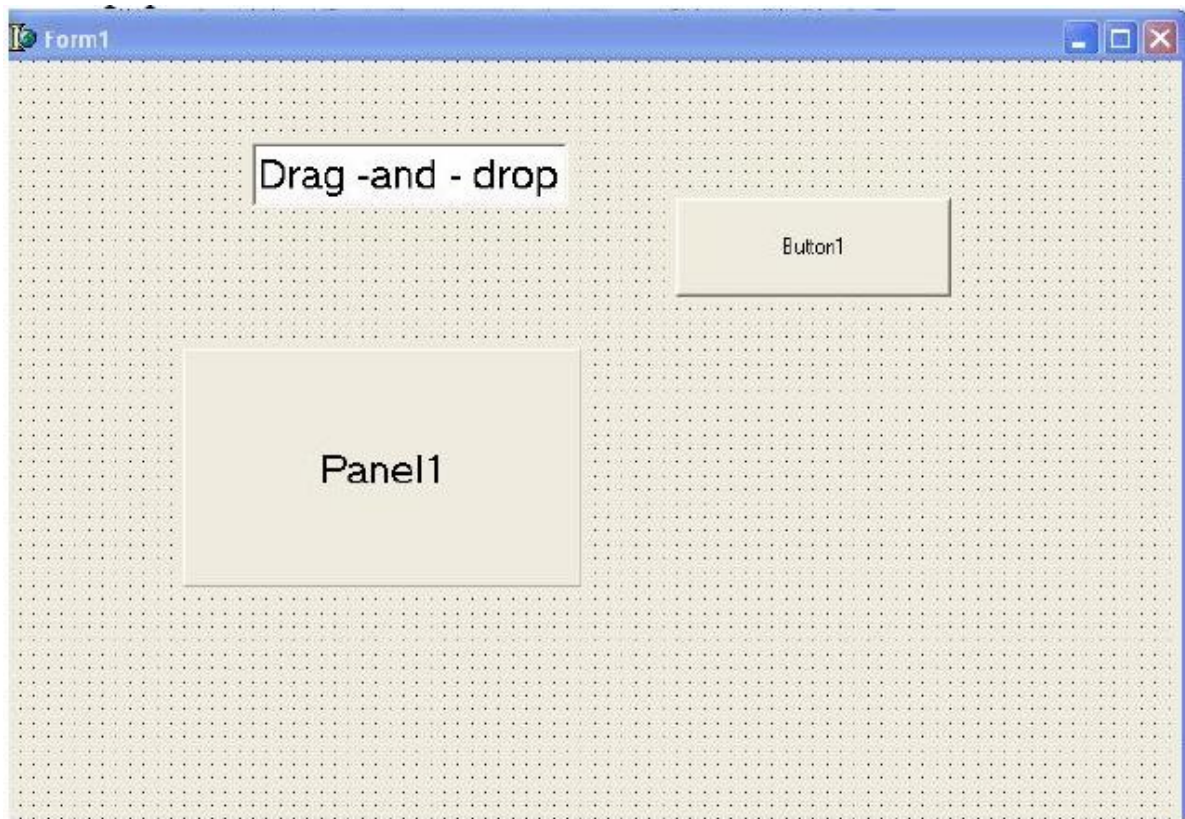
2. Відбувається обробка процесу перетягання. Для об'єктів, що можуть прийняти даний об'єкт задається подія OnDragOver. Об'єкт повинен передати об'єкту, що перетягається значення True у параметр обробника подій Accept.

3. Коли об'єкт, що перетягається відпущено, генерується подія OnDragDrop.

4. По завершенні перетягання генерується подія OnEndDrag. Необхідне для перевірки чи успішно завершилось перетягання.

*Реалізуємо механізм перетягання двома способами:*

Розмістіть на формі Edit, Panel, button.



1. Для об'єкта Edit1 створіть обробник подій OnMouseDown. У якому перевіряється

чи натиснута ліва клавіша миші, та викликається метод BeginDrag з параметром False. Це означає, що процес перетягання не почнеться поки миша не пересунеться на деяку відстань.

```
procedure TForm1.Edit1MouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin
  if button = mbLeft then
    edit1.BeginDrag(false)
  ;
end;
```

2. Для об'єкта Panel1 створіть подію OnDragOver. У якому перевіряється чи дійсно

об'єкт що перетягається є полем введення Edit1. Результат перевірки заноситься у параметр Accept.

```
procedure TForm1.Panel1DragOver(Sender, Source: TObject; X, Y: Integer;
State: TDragState; var Accept: Boolean);
begin
  Accept:=Source=Edit1;
end;
```

3. Панель повинна обробляти ще одну подію OnDragDrop, яка генерується під час

кидання об'єкту. Та відбувається запис тексту з текстового поля Edit1 у заголовок панелі Panel1.

```
procedure TForm1.Panel1DragDrop(Sender, Source: TObject; X, Y: Integer);
begin
  Panel1.Caption :=Edit1.Text;
end;
```

### **Другий приклад механізму перетягання.**

Для об'єкту Button1 створюється механізм обробки натискання кнопки миші, у якому починається процес перетягання кнопки по формі.

```
procedure TForm1.Button1MouseDown(Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer);  
begin  
    if button = mbLeft then  
        Button1.BeginDrag(false);  
end;
```

Для того, щоб форма мала право приймати кнопку вона повинна обробляти подію OnDragOver

```
procedure TForm1.FormDragOver(Sender, Source: TObject; X, Y: Integer;  
State: TDragState; var Accept: Boolean);  
begin  
    Accept:=Source is  
    TButton; end;
```

Також необхідно обробляти подію OnDragDrop. Під час обробки для вихідної кнопки (параметр Source, який приведений до типу TButton за допомогою операції as) задаються нові координати її положення на формі.

```
procedure TForm1.FormDragDrop(Sender, Source: TObject; X, Y: Integer);  
begin  
    (Source as Tbutton).Left:=X;  
    (Source as TButton).Top:=Y;  
end;
```

### **Інтерфейс приєднання Drag-and-Dock**

Механізм дозволяє динамічно переміщати елементи управління між носіями (доками) за допомогою миші (аналогічно до панелей інструментів Office).

Для включення автоматичного впровадження Drag-and-Dock необхідно в інспекторі об'єктів змінити властивість DragKind на dkDock, а властивість DragMode — на dmAutomatic.

Зараз цей елемент керування можна перетягати з одного носія - дока на інший.

Носієм інших компонентів (доком) може служити нащадок TWinControl.

Він має властивість Docksite, встановлення якого у True дозволяє перенос на нього інших компонентів. Якщо при цьому встановити властивість Autosize у True, док буде автоматично масштабуватись в залежності від того, що на ньому знаходиться.

Кожний елемент керування, що переноситься має дві події, що виникають у моменти початку та кінця переносу.

```
type  
TStartDockEvent = procedure(Sender: TObject; var DragObject: TDragDockObject) of  
object;  
TEndDragEvent = procedure(Sender, Target: TObject; X, Y: Integer) of object;
```

У першому методі Sender - це об'єкт, що переноситься, а DragObject - спеціальний об'єкт, що створюється на час переносу та вміщає його властивості. У другому Sender - це також об'єкт, що переноситься, а Target - об'єкт-док.

Док також повідомляється про події під час переносу:

*type*

```
TGetSiteInfoEvent = procedure(Sender: TObject; DockClient: TControl; var  
InfluenceRect: TRect; MousePos: TPoint; var CanDock: Boolean) of object;
```

```
TDockOverEvent = procedure(Sender: TObject; Source: TDragDockObject; X, Y:  
Integer; State: TDragState; var Accept: Boolean) of object;
```

```
TDockDropEvent = procedure(Sender: TObject; Source: TDragDockObject; X,  
Y: Integer) of object;
```

```
TUnDockEvent = procedure(Sender: TObject; Client: TControl; NewTarget:  
TWinControl; var Allow: Boolean) of object;
```

Як тільки користувач натиснув кнопку миші над об'єктом, що переноситься, всім докам розсилається подія onGetsiteinfo. З ним передаються параметри: DockClient - від об'єкта, що бажає "приземлитись" та MousePos.

У відповідь док повідомляє рішення: параметр CanDock - чи приймає компонент та influenceRect - прямокутник, що представляється.

За допомогою цієї події можна приймати лише визначені елементи керування, наприклад:

```
procedure TForm1.PanellGetSiteInfo(Sender: TObject; DockClient: TControl; var  
InfluenceRect: TRect; MousePos: TPoint; var CanDock: Boolean);
```

```
begin
```

```
if DockClient is TBitBtn  
then CanDock := False;
```

```
end;
```

Якщо є необхідність змінити панель інструментів, то необхідно створити нащадка від класу TCustomDockForm та зв'язати властивість FloatingDockSiteClass з цим новим класом:

```
TMyCustomFloatingForm =  
class(TCustomDockForm) public
```

```
constructor Create(AOwner: TComponent);  
override; end;
```

```
constructor TMyCustomFloatingForm.Create(AOwner:  
TComponent); begin
```

```
inherited Create(AOwner);  
BorderStyle := bsNone;
```

```
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);  
begin
```

```
ToolBar1.FloatingDockSiteClass := TMyCustomFloatingForm;
```

```
end;
```

В цьому прикладі несуче вікно плаваючої панелі інструментів не має заголовка.

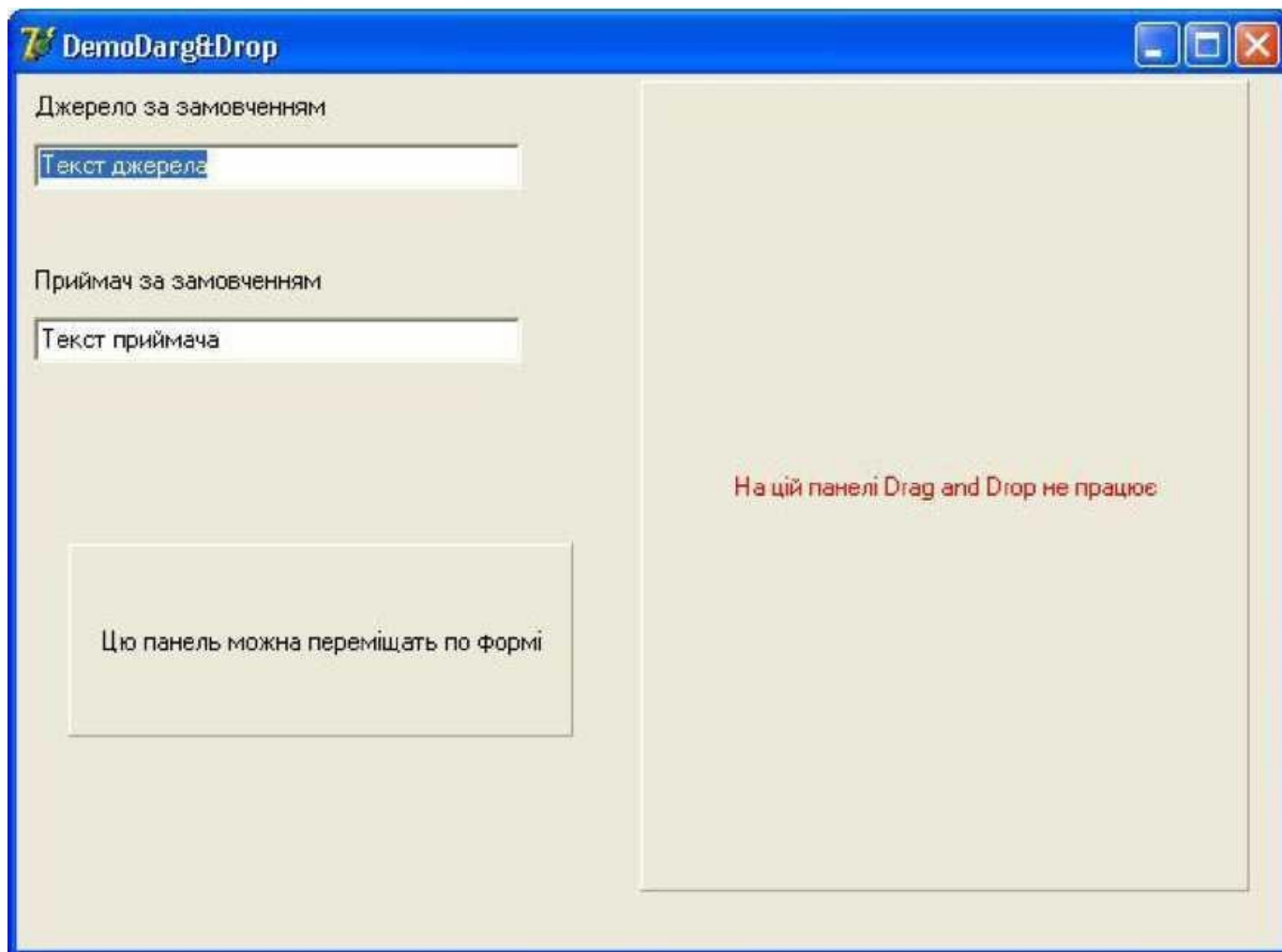
## Контрольні запитання

1. Вкажіть етапи механізму перетягання об'єктів.
2. Опишіть один механізм перетягання об'єкту по формі?

### **Порядок виконання роботи**

1. Запустити середу програмування Delphi;
2. Реалізувати наведені в практичній роботі приклади.
3. Створити власну програму з застосуванням механізму перетягування об'єктів.

**Задача 1.** На основі механізму перетягання реалізувати передачу тексту між текстовими полями та переміщення панелі по формі.



Лістинг.

*implementation*  
*n {\$R \*.DFM}*

```
procedure TForm.Edit1MouseDown(Sender: TObject; Button: TMouseButton; Shift: TShiftState;
X, Y: Integer);
begin
```

```
    if Button = mbLeft then
        TEdit(Sender).BeginDrag(True)
    ;
```

```
end;
```

```
procedure TForm.Edit2DragOver(Sender, Source: TObject; X, Y: Integer; State: TDragState; var
Accept: Boolean);
begin
```

```
    if Source is TEdit then
        Accept := True
    else
        Accept := False;
```

```
end;
```

```
procedure TForm.Edit2DragDrop(Sender, Source: TObject; X, Y: Integer);
begin
```

```
    TEdit(Sender).Text := TEdit(Source).Text;
    TEdit(Sender).SetFocus;
    TEdit(Sender).SelectAll;
```

```
end;
```

```
procedure TForm.Edit1EndDrag(Sender, Target: TObject; X, Y: Integer);
begin
```

```
    if Assigned(Target) then  
        TEdit(Sender).Text := 'Текст перенесено у ' + TEdit(Target).Name;  
end;
```

```
procedure TForm.FormDragOver(Sender, Source: TObject; X, Y: Integer; State:  
TDragState; var Accept: Boolean);  
begin  
    if Source.ClassName = 'TPanel' then Accept := True  
    else  
        Accept := False;  
end;
```

```
procedure TForm.FormDragDrop(Sender, Source: TObject; X, Y:  
Integer); begin  
    TPanel(Source).Left := X;  
    TPanel(Source).Top := Y;  
end;  
end.
```