



ПОДІЇ ОБРОБНИКИ ПОДІЙ

Техноспот, Івано-Франківськ

ПОДІЇ У ПОВСЯКДЕННОМУ ЖИТТІ

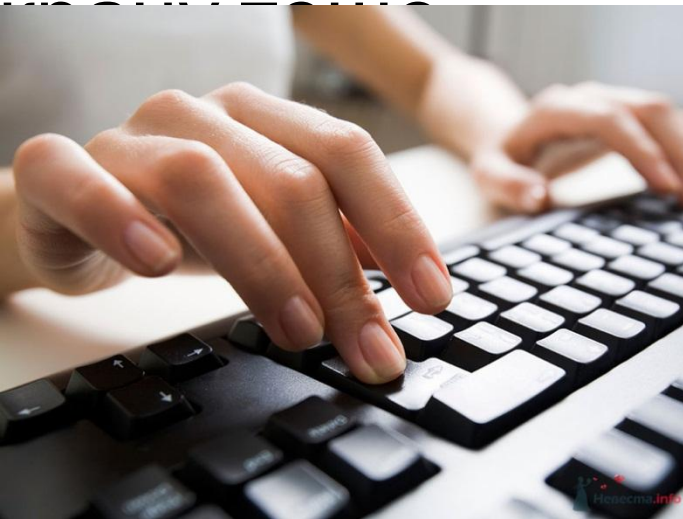
Будь-який об'єкт реальному житті змушений реагувати на ті чи інші події навколишнього світу: задзвонив телефон - людина бере трубку, пішов дощ - ми відкриваємо парасольку.



ПОДІЇ ДЛЯ ПРОГРАМИ

Програми, з якими ми працюємо, теж повинні реагувати на різні події, які виникають внаслідок певних дій користувача: натискання клавіші на клавіатурі, клацання кнопкою миші, переміщення курсору по

е



ПОДІЇ ДЛЯ ПРОГРАМИ

Середовище Lazarus надає можливість програмі, яку ми створюємо, реагувати на певні події: програміст сам вирішує, на які дії користувача повинен реагувати той чи інший компонент (об'єкт), створюючи для кожної потрібної події відповідну **процедуру реагування** на неї. Така процедура містить команди, які потрібно виконати при настанні очікуваної події, і називається **обробником вказаної події**.

ПОДІЇ ДЛЯ ПРОГРАМИ

Операційна система Windows першою дізнається про будь-яку подію, і якщо зв'язок з обробником цієї події встановлений, передає управління даним процедурі.

1

- Користувач натиснув кнопку миші

2

- Windows генерує подію **Click**

3

- Обробник події **OnClick** виконує потрібні команди

ПОДІЇ ДЛЯ ПРОГРАМИ

Наведемо приклади найчастіше використовуваних подій

Назва події	Коли виникає
<i>OnClick</i>	При клацанні по компоненту
<i>OnKeyPress</i>	При натисканні довільної клавіші
<i>OnDBClick</i>	При подвійному клацанні
<i>OnMouseMove</i>	При переміщенні курсору над компонентом
<i>OnMouseDown</i>	При натисканні кнопки миші
<i>OnMouseUp</i>	При відпусканні кнопки миші
<i>OnEnter</i>	При отриманні компонентом фокусу введення
<i>OnExit</i>	При втраті компонентом фокусу введення
<i>OnCreate</i>	Тільки для форми – при створенні форми

ОБРОБНИК ПОДІЇ

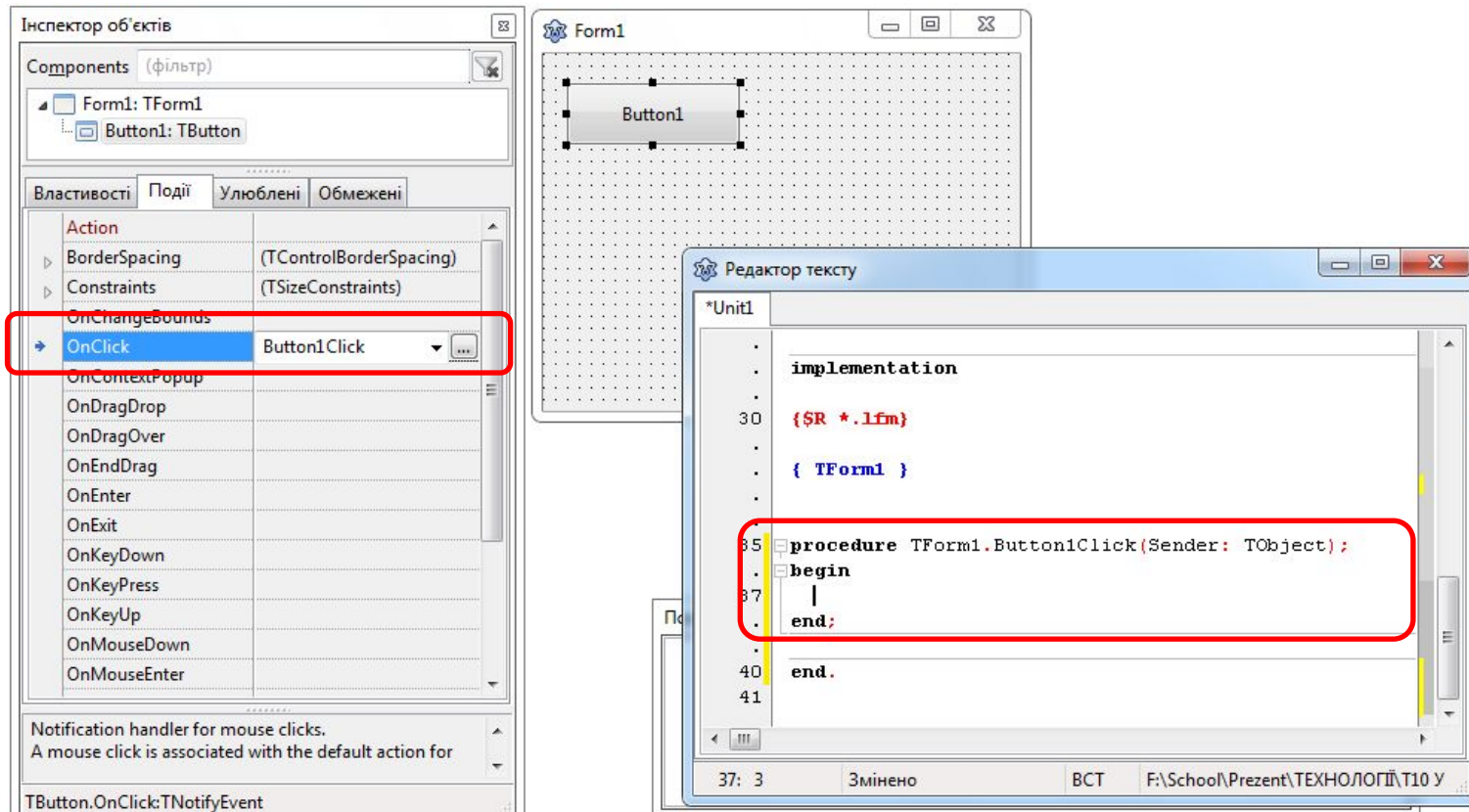
The screenshot displays the Lazarus IDE v1.2.0 interface. The main window shows a form with a button component. The Object Inspector on the left is open, and the 'Events' tab is selected, showing a list of events for the button. A red circle highlights the 'Events' tab and the list of events. A red box highlights a zoomed-in view of the 'Events' tab, showing the 'Action' event and its associated handler.

Event	Handler
Action	
BorderSpacing	(TControlBorderSpacing)
Constraints	(TSizeConstraints)
OnChangeBounds	
OnClick	
OnContextPopup	
OnDragDrop	
OnDragOver	
OnEndDrag	
OnEnter	
OnExit	
OnKeyDown	
OnKeyPress	
OnKeyUp	
OnMouseDown	
OnMouseEnter	

```
public declarations }  
  
25 Form1: TForm1;  
.  
implementation  
.  
30 {$R *.lfm}  
.  
{ TForm1 }
```

Для створення обробника подій потрібно виділити на формі потрібний компонент і в Інспекторі об'єктів перейти на вкладку «Події»

ОБРОБНИК ПОДІЇ



Після цього потрібно вибрати у списку потрібну подію і виконати подвійне клацання напроти її назви – при цьому у програмному коді згенерується заготовка процедури-обробника цієї події.

ОБРОБНИК ПОДІЇ

Для прикладу, заготовка обробника події клацання мишею по кнопці виглядає так:

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    |  
end;
```

Програмісту залишається записати потрібний програмний код (послідовність потрібних команд) між словами `begin` (початок процедури обробки події) та `end` (кінець процедури обробки події).

ОБРОБНИК ПОДІЇ

Для прикладу наведемо обробник події **Button1.Click** (натискання кнопки **Button1**), в результаті чого форма змінить свій колір на червоний і набуде розмірів 400*300 пікселів.

```
35 procedure TForm1.Button1Click(Sender: TObject);  
begin  
    Form1.Color:=clRed;  
    Form1.Width:=400;  
    Form1.Height:=300;  
40 end;  
end.
```

Нагадаємо, що всі ці команди будуть виконані під час роботи створеної програми при клацанні користувачем кнопки 1 на формі.