

Створення найпростішої Windows-програми

Практична робота № 1

Створення найпростішої Windows-програми



Мета роботи: набути практичних навичок створення простих додатків.

Завдання. Написати програму-підказку для навчання англійської мови. Для цього на форму виводиться комплект картинок; унаслідок клацання мишею на кожній із них у нижній частині форми виводиться підпис — назва зображеного на картинці предмета англійською мовою (рис. 2.13).

Хід роботи

1. Запустимо середовище Delphi та створимо новий додаток.
2. На заготовку форми, що з'являється на екрані після запуску Delphi (або внаслідок вибору через основне меню File → New → Application), помістимо 12 об'єктів Image. Відеоконponent Image, призначений для виведення на форму малюнків у форматах BMP, JPG, EMF, WMF, ICO, розміщується на другій сторінці списку відеоконponentів і має іконку .

Виведення 12 малюнків (Image) однакових розмірів — доволі трудомістка операція. Простіше вивести один Image, скопіювати його натисканням



Рис. 2.13. Програма, що виводить англійські назви предметів на картинках

комбінації клавіш **Ctrl + C**, після чого вивести на форму ще 11 малюнків натисканням клавіш **Ctrl + V**. При цьому середовище Delphi саме нумерує відеоконпоненти **Image**, **Image3** і т. д. аж до **Image12**. Тепер по черзі клацаємо на кожному компоненті **Image** і налаштуємо програму на виведення відповідної картинки. Для цього обираємо для компонента **Image** в **Object Inspector** рядок **Picture**. Клацанням мишею на кнопці з трьома крапками напроти назви властивості **Picture** відкриваємо вікно завантаження файлу (рис. 2.14).

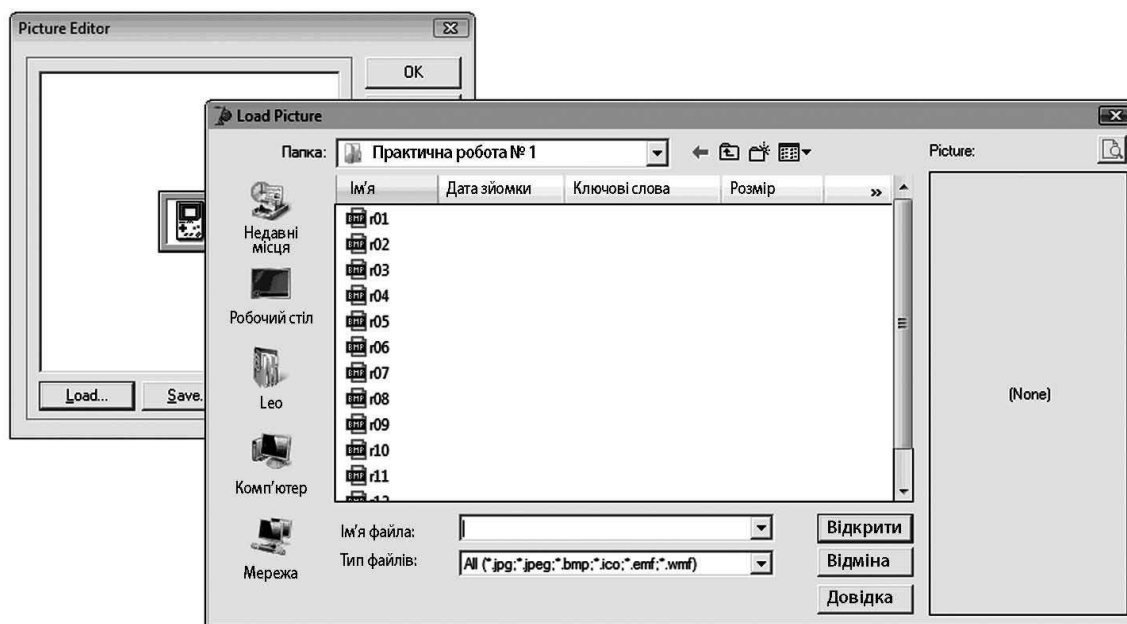


Рис. 2.14. Завантаження малюнка з файлу в компонент **Image** клацанням на кнопці **Load**

3. Виведення малюнка в об'єкти **Image** доведеться налаштувати 12 разів — для кожного об'єкта є свій діалог на вибір малюнка. Завдання можна полегшити, якщо всі файли картинок зібрати в одній папці, після чого зберегти в ній створюваний проект. У цьому випадку в діалозі не доведеться шукати малюнки, переглядаючи різні папки. Якщо розмір малюнка не збігається з розміром контурів компонента **Image**, рисунок можна розтягти, встановивши у властивості **Stretch** (розтягнення) значення **true**.
4. У нижній частині форми помістимо компонент **Label** (іконка **A** на першій закладці **Standard**). За замовчуванням у компонента **Label** розмір шрифту нехай дорівнює 8, букви світлого написання. Через **Object Inspector** настроїмо наш рядок виведення тексту на більші й напівжирні букви: властивість **Font** → **Size** встановимо 12, а властивості **Font** → **Style** → **fsBold** присвоїмо значення **true**.
5. Тепер напишемо текст програми для обробників подій «натиснута клавіша миші під час перебування курсора над об'єктом». Почнемо з першого малюнка — **Image1**. На нього виведено зображення сиру («сир» англійською «cheese»). Вибираємо в **Object Inspector** для компонента **Image1** подію **OnMouseDown** і клацаємо мишею

на вільному полі. Автоматично створюється оголошення методу оброблення події, між **begin** і **end** вписуємо всього один рядок програмного коду:

```
-----  
procedure TForm1.Image1MouseDown(Sender: TObject;  
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);  
begin  
    Label1.Caption:='Cheese';  
end;  
-----
```

6. Очевидно, що цей рядок можна скопіювати і вставити в усі обробники натисканням клавіші миші на решті 11 малюнках, міняючи тільки слово Cheese на інші англійські слова. Для об'єктів на рис. 2.14 це будуть Calculator, Candle, Bouquet, Car, Pumpkin, Palm-tree, Plane, Jeep, Dices, Anchor, Hare (написи в операторах присвоювання виведеного рядка, природно, збігаються з послідовністю завантажених у компоненти Image картинок).
7. Збережемо отриманий додаток на диску й запусимо його на виконання.