

Тема 4. Основи об'єктно-орієнтованого програмування

План

1. Суть об'єктно-орієнтованого підходу в програмуванні
2. Робота з моделлю об'єктів Microsoft Excel на прикладі діалогових вікон

4.1. Суть об'єктно-орієнтованого підходу в програмуванні

Visual Basic на базі пакету Microsoft Office – об'єктно-орієнтована мова програмування. Це дозволяє спростити процес написання, сприйняття тексту процедур й робить їх більш ефективними.

Об'єктно-орієнтований підхід передбачає наявність ієрархії об'єктів, з якими взаємодіє розробник. Кожний об'єкт характеризується наступними категоріями:

1. Події;
2. Властивості;
3. Методи.

Сутність об'єктно-орієнтованого програмування полягає в налаштуванні й роботі з зазначеними категоріями об'єктів. Розглянемо їх більш детально.

1. Події. В житті кожного об'єкта відбуваються події, на які він має можливість належним чином реагувати. Приклади подій:

- Activate – подія відбувається в момент активації об'єкта на екрані (відкриття діалогового вікна);
- Click – подія відбувається, коли користувач клікає над об'єктом лівою кнопкою миші один раз;
- DblClick – подія відбувається, коли користувач робить над об'єктом подвійний клік лівою кнопкою миші;
- MouseDown – подія відбувається в момент натискання лівої, або правої кнопки миші над об'єктом;
- MouseUp – подія відбувається в момент відпускання лівої, або правої кнопки миші над об'єктом;
- MouseMove – подія відбувається в момент наведення курсору миші над об'єктом;
- Enter – подія відбувається в той момент, коли об'єкт отримує фокус (стає активним);
- Exit – подія відбувається в той момент, коли об'єкт втрачає фокус.

Кожний об'єкт має свою індивідуальну множину подій. Відповідна процедура-обробник кожної події оформляється у вигляді процедури, з назвою: НазваОб'єкту_Подія(), наприклад:

```
Private Sub UserForm1_Activate()  
    ...  
End Sub
```

Програмний код, написаний в межах цієї процедури буде виконуватись кожного разу, коли з об'єктом UserForm1 буде відбуватись подія Activate.

2. Властивості. Кожний об'єкт має свою унікальну множину властивостей, які можуть приймати різні значення й визначають його назву, зовнішній вигляд, розміри, місце розташування, тип шрифту та інше.

Налаштовувати значення властивостей об'єктів можна на етапі розробки додатку, або програмно, в процесі виконання коду.

Для перегляду, або редагування значень властивостей об'єктів в процесі розробки програмного коду, використовується вікно властивостей VBA, рис. 4.1.

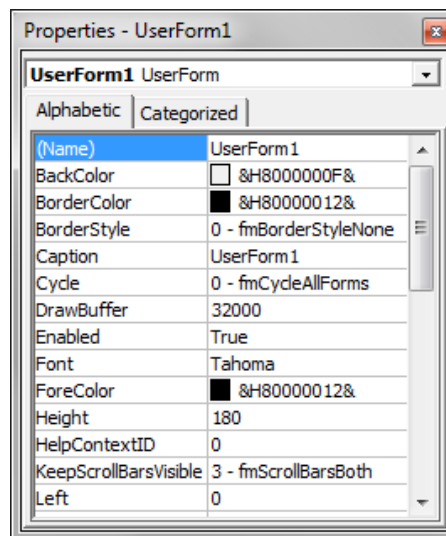


Рис. 4.1. Вікно властивостей редактору коду VBA

Наприклад, у вікні властивостей можна вказати фон для форми (кольор), або змінити назву аркуша робочої книги Excel.

На рис. 4.1 зображено вікно властивостей для об'єкту UserForm1 (користувацьке діалогове вікно).

У вікні властивостей відображаються властивості поточного об'єкту. Для відображення властивостей будь-якого іншого об'єкту, слід обрати його зі списку, розташованого у верхній частині цього вікна (в даний момент вибраний об'єкт UserForm1).

Основна частина вікна властивостей складається з двох колонок. Перелік властивостей виводиться в лівому стовпці а їхні значення – у правому. Щоб змінити значення певної властивості, обираємо її з лівого стовпця й вносимо відповідні правки в колонці справа.

Встановлення, або зчитування властивостей об'єкту програмним шляхом, виконується наступним чином:

```
НазваОб'єкту.Властивість = Значення  
НазваЗмінної = НазваОб'єкту.Властивість
```

Приклади властивостей об'єктів:

- Name – назва об'єкту. Використовується при зверненні до нього програмним шляхом;
- Caption – заголовок об'єкту, його підпис (напис на кнопці, або в заголовку діалогового вікна);
- Enabled – показує стан об'єкта «активний / неактивний». Приймає значення «True», або «False»;
- Height – висота об'єкту в пікселях;
- Width – ширина об'єкту в пікселях;
- Font – шрифт, за допомогою якого виводиться текст в межах даного об'єкту;
- Value – числове значення, яке привласнюється об'єкту;
- Text – текстове значення, яке привласнюється об'єкту.

3. Методи. Більшість об'єктів можуть виконувати певні дії, які називаються методами.

Множина методів кожного з об'єктів є настільки індивідуальною, що немає сенсу говорити про найбільш поширені методи.

Наказати об'єкту виконати певну дію можна лише програмним шляхом, використовуючи синтаксис:

```
НазваОб'єкту.Метод
```

Робоча книга Microsoft Excel для розробника представляє собою безліч об'єктів. Тому, перш ніж приступати до написання додатків, які автоматизують роботу користувача, необхідно ознайомитись з моделлю об'єктів, їхньою ієрархією. Ієрархія об'єктів визначає взаємозв'язок між ними й має деревоподібну структуру.

Щоб використовувати властивості, або методи конкретного об'єкту, найчастіше доводиться вказувати його положення в структурі об'єктів робочої книги. Наприклад, якщо «Об'єкт2» за ієрархією є на рівень нижчим, ніж «Об'єкт1», то для звернення до його властивостей використовується синтаксис:

```
Об'єкт1.Об'єкт2.Властивість = Значення
```

Наприклад, як ми раніше звертались до комірок робочого аркуша Excel:

```
Worksheets("Аркуш1").Cells(3,4).Value = 5
```

Однак, не в усіх випадках потрібно явно вказувати місце елемента в ієрархії об'єктів. Якщо по контексту ясно, про що йде мова, то можна використовувати об'єкти, не вказуючи їхню приналежність рівнів ієрархії:

```
Range("A5").Value = 10
```

4.2. Робота з моделлю об'єктів Microsoft Excel на прикладі діалогових вікон

Досить часто, проекти VBA потребують додавання користувацьких діалогових вікон (форм), які полегшують взаємодію користувачів з даними.

Форма – це те ж саме, що й будь-яке діалогове вікно операційної системи Windows. Єдина відмінність полягає в тому, що розробник може самостійно створювати форми під індивідуальні потреби чи поставлене завдання.

В редакторі VBA можна додавати довільну кількість форм. Для цього необхідно виконати наступні дії:

1. Відкрити редактор VBA;
2. Вибрати команду головного меню Insert, UserForm. На екрані з'являється порожня форма й панель елементів керування;
3. Вибрати команду головного меню View / Properties Window. Виводиться вікно властивостей;
4. Задати властивості нової форми, наприклад, її назву (властивість Name).

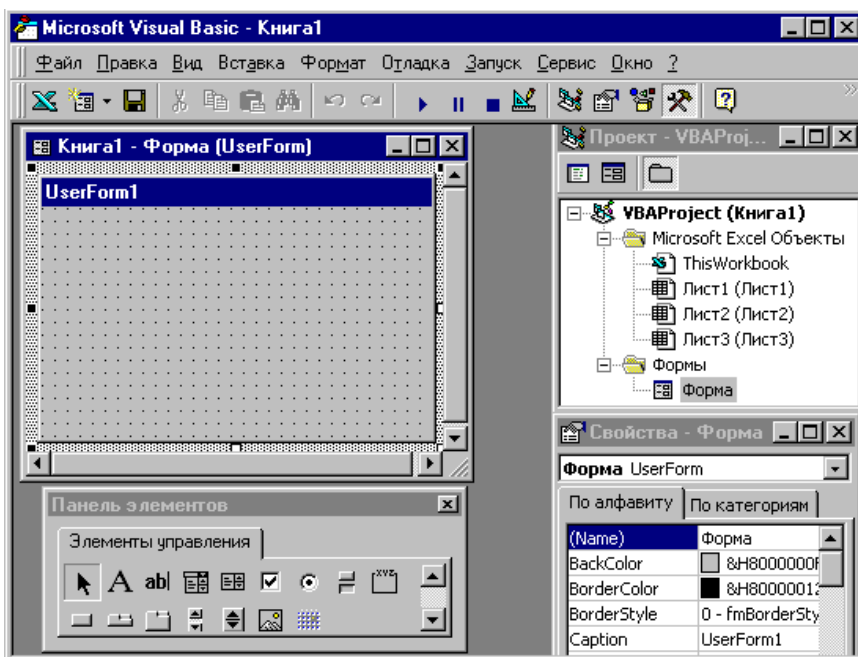


Рис. 4.2. Створення діалогового вікна у VBA

Панель елементів керування дозволяє розмістити ряд елементів управління на формі:

1. Виберіть форму, на яку слід додати елементи керування. При необхідності введіть набір інструментів.
2. Натисніть на панелі інструментів кнопку необхідного елемента, а потім клікніть по будь-якому місцю на формі. На формі з'явиться новий елемент керування.
3. Змініть розміри елемента керування, перетягнувши його рамку. Те ж саме можна зробити й з формою.

Елемент керування TextBox дозволяє розмістити на формі текстове поле, для введення даних користувачем, які потім можна використовувати в програмі.

Після створення елемента керування найкраще одразу присвоїти йому нову назву (властивість Name). Якщо назву поля не вказувати, то використовується назва за замовчуванням.

Для запису й зчитування вмісту текстового поля використовується властивість Value. Наприклад, в поле TxtControl запишемо значення "abcdef":

```
UserForm1.TxtControl.Value = "abcdef"
```

В наступному прикладі до текстового поля запишемо число 100:

```
UserForm1.TxtControl.Value = 100
```

Для зчитування значення даного TextBox, можна використовувати наступний фрагмент:

```
a = UserForm1.txtControl.Value
```

Елемент керування Label використовується для виведення підписів на формі. Найбільш важливою його властивістю є Caption, що містить текст підпису. Властивість Caption можна задавати у вікні властивостей, або в тексті програми:

```
UserForm1.labControl.Caption = "Адреса"
```

Елемент керування CommandButton ініціює виконання певних дій по натисненню на кнопку. За допомогою властивості Caption є можливість задати надпис, який виводиться на кнопці. З натисканням кнопки можна пов'язати виконання певної дії, якщо призначити процедуру обробки її події Click.

Елемент керування OptionButton (перемикач) використовується для надання вибору одного варіанта з багатьох. В один й той самий момент у групі перемикачів може бути обраний тільки один перемикач. Скасування вибору одного перемикача при виборі іншого відбувається автоматично.

Для визначення вибраного перемикача використовується властивість Value. Якщо значення даної властивості дорівнює True, то даний перемикач включений, якщо False – вимкнений.

Крім перерахованих, існують й інші елементи керування. Вичерпну інформацію з них можна отримати в довідковій системі по VBA.

Після розміщення, вирівнювання та завдання розмірів й властивостей елементів керування, слід пов'язати їх з певним програмним кодом:

- запуск якоїсь дії при натисненні на кнопку;
- збереження даних форми на робочому аркуші Excel, або зчитування даних з аркуша на форму.

Для виконання даної операції, слід виконати подвійний клік по елементу керування на формі. З'являється вікно модуля для вибраного об'єкту. З випадаючого списку зверху справа слід обрати подію, для якої буде створюватись процедура.

Для запуску форми на виконання, необхідно натиснути клавішу <F5>, або скористатись командою меню Run, Run Sub/Forms.