

Тема 5. Організація вводу/виводу в діалогових вікнах

1. Підключення бібліотек для вводу/виводу в діалогових вікнах.
2. Організація вводу в діалогових вікнах.
3. Організація виводу в діалогових вікнах.

1. В консольному додатку по замовчуванню ввід/вивід виконується в Console за допомогою методів WriteLine, ReadLine. Щоб вводити/виводити в діалогових вікнах, спочатку необхідно підключити бібліотеки (.dll), де містяться відповідні методи, а потім вже з них відкривати простори імен і використовувати класи.

Для організації виводу ми використовуватимемо бібліотеку System.Windows.Forms, а для організації вводу – Microsoft.VisualBasic.

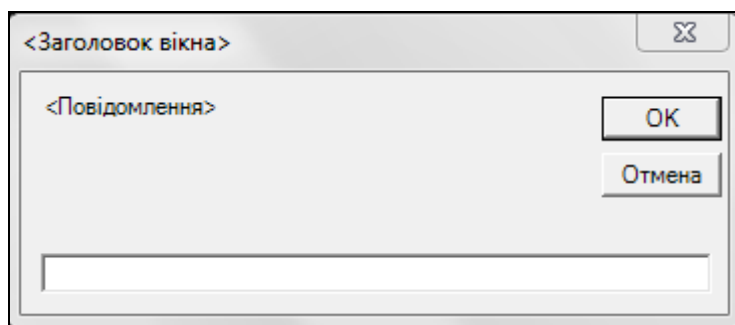
Для підключення бібліотек необхідно у вікні рішення віднайти групу посилань References і в її контекстному меню обрати пункт Додати посилку, віднайти посилання на потрібні бібліотеки та встановити біля них прапорці і натиснути ОК.

2. Для організації вводу в діалогових вікнах ми використовуватимемо метод Interaction.InputBox, який описаний у просторі імен Microsoft.VisualBasic.

Синтаксис методу:

```
string Interaction.InputBox(string <повідомлення>, [string  
    <заголовок вікна>], [string <значення на замовчування>])
```

Цей метод виводить на екран діалогове вікно з вказаним повідомленням і заголовком, рядком вводу і кнопками ОК та Cancel (Отмена).



Обов'язковим параметром методу є лише перший, але хорошим стилем програмування передбачається задання значень по замовчуванню.

Після натиснення ОК метод повертає в програму введений рядок, якщо ж натиснути Cancel, або закрити вікно, то в програму повернеться порожній рядок. Надалі користувач сам має перетворити введений рядок до потрібного типу даних.

Наприклад, введемо радіус зовнішнього кола в попередній програмі:

```
R1 = 5;  
S = Interaction.InputBox("Введіть радіус зовнішнього кола",  
    "Введення", R1.ToString());  
R1 = Convert.ToDouble(S);
```

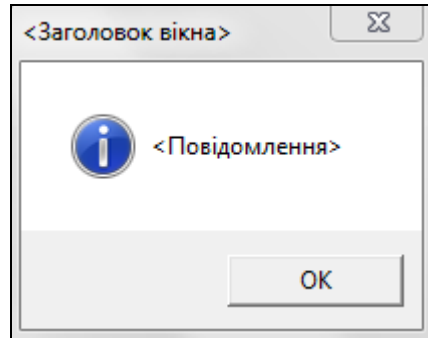
3. Для організації виводу скористаємося методом Show класу MessageBox. Для цього методу будемо використовувати максимум 4 аргументи.

Синтаксис методу:

```
DialogResult MessageBox.Show(string <повідомлення>, string  
    <заголовок вікна>, MessageBoxButtons.<набір кнопок>,  
    MessageBoxIcon.<константа значка>)
```

де в перелічуванні MessageBoxButtons можливий вибір однієї з констант: AbortRetryIgnore, OK, OKCancel, RetryCancel, YesNo, YesNoCancel, а в перелічуванні MessageBoxIcon – однієї з констант: Asterisk, Information

(інформаційні повідомлення), Stop, Error, Hand (критичні повідомлення), Warning, Exclamation (знак оклику), None, Question. Результатом виконання функції є константа натиснутої користувачем кнопки з перелічування DialogResult.



Методи `InputBox` та `Show` можуть використовуватися з різною кількістю аргументів. Всі ці варіанти реалізовані в бібліотеках. Методи з однаковою назвою, але різною кількістю аргументів називаються **перезавантажувальними**.

Приклад:

Обчисліть периметр і площу прямокутного трикутника, якщо відомі гіпотенуза і один з його катетів.

Для організації вводу/виводу після підключення бібліотек в модулі програми після стандартних відкриваємо додаткові простори імен:

```
using System.Windows.Forms;
using Microsoft.VisualBasic;
```

Текст програми в методі `Main`:

```
Double a, b, c, P, S;
String s;
c = 5;
a = 3;
s = Interaction.InputBox("Введіть гіпотенузу", "Введення",
    c.ToString());
c = Convert.ToDouble(s);
s = Interaction.InputBox("Введіть довжину катета",
    "Введення", a.ToString());
a = Convert.ToDouble(s);
b = Math.Sqrt(c*c-a*a);
P = a+b+c;
S = (a+b)/2;
MessageBox.Show("Якщо гіпотенуза рівна " + c.ToString() +
    ", а один з катетів " + a.ToString() +
    ", то другий катет становитиме " + b.ToString() +
    ", периметр трикутника " + P.ToString() +
    ", а площа " + S.ToString());
```

При вводі можна сформувати заголовок вікна, кнопки та значки. Для цього після повідомлення в дужках потрібно записати:

```
, "Результати обчислень", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
```

Повідомлення, яке виводиться в діалоговому вікні, можна розбивати на декілька фізичних рядків. Для цього в текст додається спецсимвол переводу рядка чи повернення каретки `Strings.Chr(13)` або `Strings.Chr(10)`.