ЛАБОРАТОРНА РОБОТА №7

Тема: Використання функцій.

Мета роботи

Дослідити принцип роботи, призначення, форму запису та особливості вживання процедур.

Постановка завдання

Розробити проект з інтерфейсом користувача, використовуючи необхідні компоненти та оброблювач подій мовою C#.

Теоретичні відомості

1. Призначення

З функціями Ви вже мали справу при створенні оброблювача події. Застосовують функцію або підпрограму (що теж саме) коли потрібно декілька разів виконати один і той самий шматок коду. Таким чином код, а разом з ним і виконуваний файл, мають менший розмір. Окрім зменшення розміру коду, функції дозволяють логічно його відокремити, що полегшує читання та редагування програми.

2. Оголошення функції

Оголошення функції починається з типу, який вона повертає, після закінчення її виконання. Далі ім’я функції та список параметрів у дужках.

```
тип_0 Ім’я_Функції(тип_1 параметр_1, … тип_n параметр_n)
{
    // тіло функції, у якому повинен на всі гілках коду
    // бути оператор "return значення_типу__тип_0;"
}
```

Якщо функція не повертає ніякого значення – таку функцію називають процедурою – то замість тип_0 треба писати void, тоді повертати значення за допомогою return необов’язково (у цьому випадку просто завершується виконання функції без повертання будь-яких значень).

3. Оголошення параметрів функції

Для звичайних змінних числового типу параметр у функцію передається по значенню. Тому якщо таким чином потрібно змінити значення декількох змінних, то нічого не вийде. Для цього необхідно передати параметр по адресу (посиланню) на зміну. Для цього є ключове слово out. Наприклад: `out int p`. Об’єкти типу за посиланням завжди передаються у функцію за посиланням (масиви, строки та всі інші класи).

Функція також може параметри за замовчуванням. Тобто якщо функція має один звичайний і один параметр за замовчуванням, то при її визові з одним параметром у параметр за замовчування буде автоматично підставлятися значення яке вказано.

Якщо кількість параметрів спочатку не відома, то можна оголосити такий параметр: `params тип[] ім'я_масиву_з_параметрами`.

Приклади виконання завдань

Приклад 1

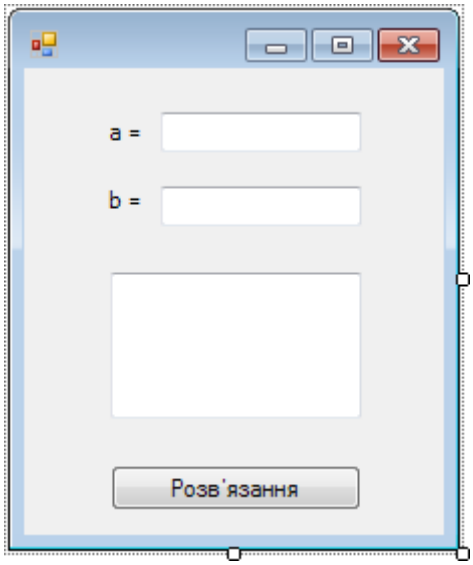
Обчислити значення наступного виразу:

$z = f(sina, b) + f(cosa, b) + f(sina - cosa, b^2 - 1) + f(cosa, b^2)$, де

$$f(a,b) = \begin{cases} a + sinb, & \text{якщо } a > 0 \\ a + b, & \text{якщо } a \le 0 \end{cases}$$

Розв'язання

Створимо форму:



Код програми:

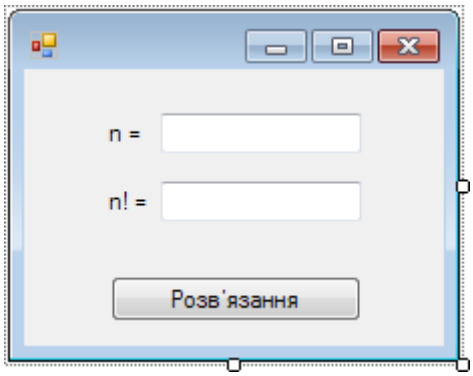
```
double f(double a, double b)
{
    return (a > 0) ? a + Math.Sin(b) : a + b;
}
// Оброблювач події натискання на кнопку "Розв'язання"
private void button1_Click(object sender, EventArgs e)
{
    double a = Convert.ToDouble(textBox1.Text),
           b = Convert.ToDouble(textBox2.Text),
           sina = Math.Sin(a),
           cosa = Math.Cos(a);
    textBox3.Text = "Результат:\n";
    textBox3.AppendText((
        f(sina, b) + f(cosa, b) +
        f(sina - cosa, b * b - 1) + f(cosa, b * b)
        ).ToString());
}
```

Приклад 2

Обчислити $n = \begin{cases} x!, & x \geq 0 \\ x^x, & x < 0 \end{cases} \quad x \in Z$

Розв’язання

Створити форму:



Написати код програми:

```
void procedure(sbyte n, out double r)
{
    if (n >= 0)
    {
        r = 1;
        for (sbyte i = n; i > 1; i--) r *= i;
    }
    else r = Math.Pow(n, n);
}
// Оброблювач події натискання на кнопку "Розв’язання"
private void button1_Click(object sender, EventArgs e)
{
    double r;
    procedure(Convert.ToSByte(textBox1.Text), out r);
    textBox2.Text = r.ToString();
}
```