

Лабораторна робота №6. Windows Forms. Елемент PictureBox

Тема: Елемент PictureBox.

Мета заняття: навчитись працювати з елементом PictureBox, закріпити навички роботи з елементами TextBox, ListBox у середовищі Microsoft Visual Studio

1. Теоретичні відомості

Елемент PictureBox призначений для показу зображень. Він дозволяє відобразити файли у форматі bmp, jpg, gif, а також метафайли зображень та іконки. Щоб вибрати зображення PictureBox можна використовувати ряд властивостей:

- **Image** : встановлює об'єкт типу Image
- **ImageLocation** : встановлює шлях до зображення на диску або в інтернеті
- **InitialImage** : деяке початкове зображення, яке буде відображатися під час завантаження головного зображення, яке зберігається у властивості Image
- **ErrorImage** : зображення, яке відображається, якщо основне зображення не вдалося завантажити в PictureBox

Щоб встановити зображення в Visual Studio, треба в панелі властивостей PictureBox вибрати властивість Image. У цьому випадку нам відкриється вікно імпорту зображення в проект, де власне і зможемо вибрати потрібне зображення на комп'ютері та встановити його для PictureBox (рис. 1.1):

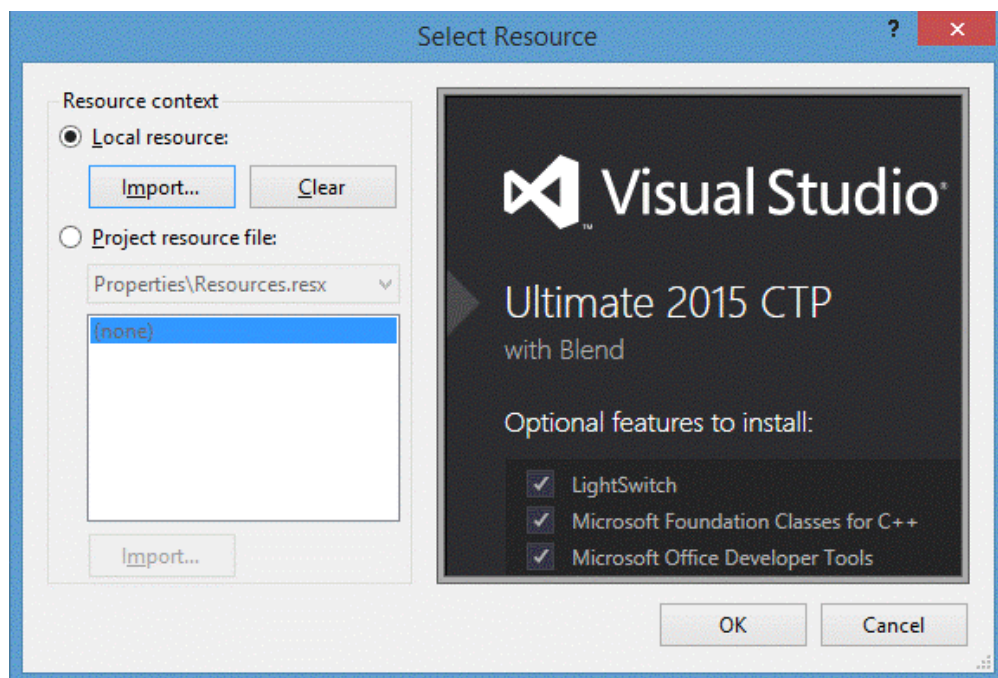


Рисунок 1.1 – Вікно імпорту зображення для PictureBox

Або можна завантажити зображення в кодї:

```
pictureBox1.Image = Image.FromFile("C:\Users\Eugene\Pictures\12.jpg");
```

Щоб вибрати зображення PictureBox використовується властивість **SizeMode**, яка набуває таких значень (рис. 1.2):

- *Normal*: Зображення позиціонується в лівому верхньому кутку PictureBox, та розмір зображення не змінюється. Якщо PictureBox більше розмірів зображення, то праворуч та знизу з'являється пусте місце, якщо менше – то зображення обрізається;
- *StretchImage*: Зображення розтягується або стискається так щоб вміститися по всій ширині та висоті елемента PictureBox;
- *AutoSize*: Елемент PictureBox автоматично розтягується, підлаштовуючись під розміри зображення;
- *CenterImage*: Якщо PictureBox менше зображення, то зображення обрізається по краях і виводиться тільки його центральна частина. Якщо ж PictureBox більше зображення, то воно позиціонується по центру;
- *Zoom*: Зображення підлаштовується під розміри PictureBox, зберігаючи при цьому пропорції.



Рисунок 1.2 – Властивість SizeMode елемента PictureBox

Для роботи з елементом PictureBox використовують такі події

- C# PictureBox Click Events
- C# PictureBox DoubleClick Events
- C# PictureBox MouseEnter Events
- C# PictureBox MouseHover Events
- C# PictureBox MouseLeave Events

Подія Click відбудеться, коли користувач натисне на PictureBox. Наприклад:

```
private void pictureBox1_Click(object sender, EventArgs e)
{
    MessageBox.Show("You Just Click On the PictureBox");
}
```

Подія DoubleClick відбудеться, коли користувач зробить подвійне натиснення на PictureBox. Обробник події MouseEnter використовується для виконання будь-яких дій, коли курсор миші знаходиться на елементі PictureBox.

Подія MouseHover відбувається кожного разу, коли користувач на деякий час наводить курсор миші на PictureBox. Існує дуже мала різниця між MouseEnter та MouseHover Event обробником. Подія MouseLeave виконується щоразу, коли курсор миші залишає межі PictureBox або видиму область.

```
private void pictureBox1_MouseHover(object sender, EventArgs e)
{
    pictureBox1.Image = Image.FromFile("C: \\Users\\Pictures\\pic1.jpg");
}

private void pictureBox1_MouseLeave(object sender, EventArgs e)
{
    pictureBox1.Image = Image.FromFile("C: \\Users\\Pictures\\pic2.jpg");
}
```

2. Практичне завдання



У процесі виконання завдань лабораторної роботи необхідно формувати набори тестових даних для перевірки правильності виконання програмного коду. Створений код і результати перевірки його роботи потрібно помістити у звіт. Тестувати роботу програми рекомендується після додання чи зміни кожного оператора виведення.

Завдання 1. Створити проект для розв'язання задачі.

Скласти програму для обчислення N-го числа Фібоначчі.

Числами Фібоначчі називають числа, які знаходять за таким правилом:

$$F_1 = F_2 = 1; \quad F_n = F_{n-1} + F_{n-2}.$$

Для обчислення N-го числа виконується така послідовність дій:

- 1) виділяються змінні **A** і **B** для зберігання двох чисел F_{n-1} , F_{n-2} ;
- 2) сума чисел $A + B$ заноситься в змінну **C**;
- 3) на наступній ітерації циклу:

$B = F_{n-1}$ стає (F_{n-2}) -м членом ряду, тому $A = B$;

$C = F_n$ стає (F_{n-1}) -м членом, тому $B = C$.

$F_1 = F_2 = 1$, тому обчислення починається з $N = 3$.

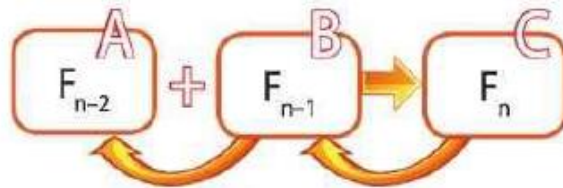


Рисунок 2.1 – Алгоритм обчислення N-го числа Фібоначчі

1. Створіть новий проект. Розробіть інтерфейс програми згідно з рис. 2.2. Додаємо на форму елементи *TextBox1*, *Label1*, *Button1*, *TextBox2*, *Label2*.

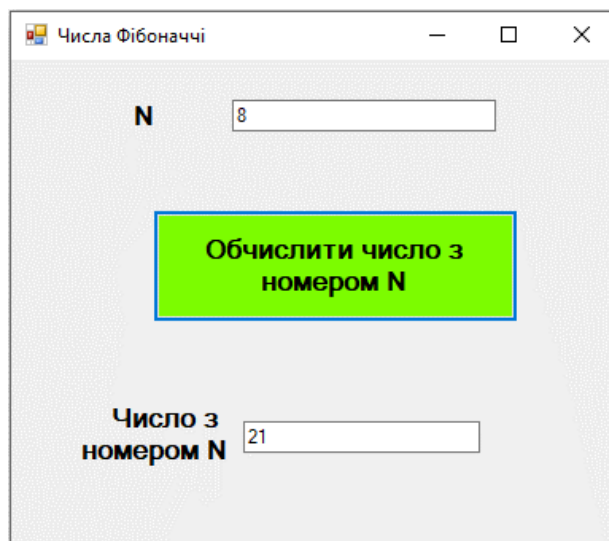


Рисунок 2.2 – Зовнішній вигляд форми для обчислення N-го числа Фібоначчі

2. Опишіть змінні, необхідні для реалізації завдання:

- n – змінна цілого типу для збереження номера елемента послідовності, який потрібно знайти;
- i – змінна цілого типу для збереження номера поточного елемента послідовності;
- a, b, c – змінні цілого типу для збереження $(i-1)$ -го, $(i-2)$ -го, i -го елементів послідовності при черговій ітерації циклу.

3. Створіть обробник події *Click* для кнопки *Button1*. Запишіть програмний код для обчислення N перших чисел Фібоначчі.

```
private void button1_Click(object sender, EventArgs e)
{
    n = Convert.ToInt16(textBox1.Text);
    a = 1; b = 1; i = 2;
    while (i < n)
    {
        i++;
        c = a + b;
        a = b; b = c;
    }
    textBox2.Text = Convert.ToString(c);
}
```

4. Випробуйте проект. Додайте на форму елемент *TextBox3*. Доповніть програмний код операторами для обчислення суми N перших чисел Фібоначчі і виведення результату до текстового поля *TextBox3*. Перевірте дію кнопки.

5. Додайте на форму елемент *PictureBox* з зображенням чисел Фібоначчі

Завдання 2. Створити проект для розв'язання задачі.

Скласти програму для розв'язання старовинної задачі. Плата за одного бика – 20 карбованців, за корову – 10 карбованців, за теля – 1 карбованців Скільки можна купити биків, корів і телят, якщо на 200 карбованців треба купити 100 голів худоби?

1. Створіть новий проект. Змініть заголовок форми на «Старовинна задача». Додайте на форму елементи *Label1*, *TextBox1*, *ListBox*, *Button1* (рис. 2.3).

2. Позначимо літерою b кількість биків; k – кількість корів; t – кількість телят. Загальна кількість голів дорівнює 100:

$$b + k + t = 100.$$

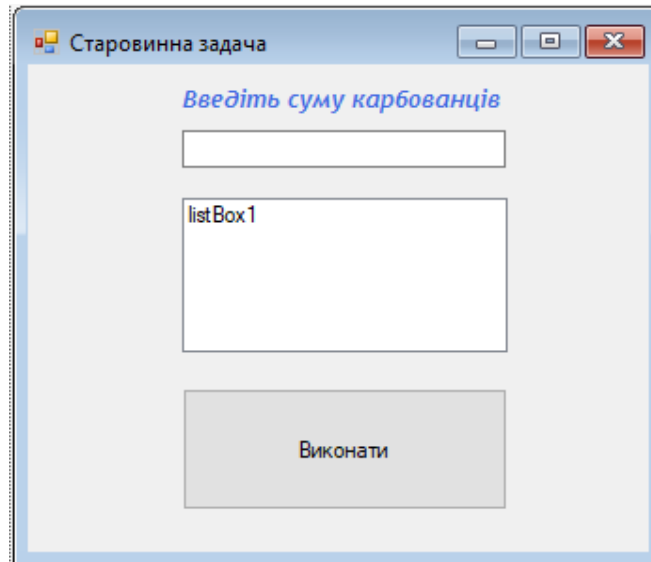


Рисунок 2.3 – Зовнішній вигляд форми для завдання 2

За биків заплатили $20b$ карбованців, за корів – $10k$ карбованців, за телят – t карбованців, отже,

$$20b + 10k + t = 200.$$

На 200 карбованців можна купити:

- не більше 10 биків, тобто $0 \leq b \leq 10$;
- не більше 20 корів, тобто $0 \leq k \leq 20$;
- не більше 200 телят, тобто $0 \leq t \leq 200$.

Таким чином, необхідно перебрати всі можливі значення змінних b , k , t і вивести в поле *ListBox* той набір значень, для яких виконується умова

$$(20 * b + 10 * k + t = 200) \text{ And } (b + k + t = 100).$$

3. Створіть процедуру обробки події Click для кнопки. Запишіть програмний код:

```
private void button1_Click(object sender, EventArgs e)
{
    int S = Convert.ToInt32(textBox1.Text);
    int N = 100;
    listBox1.Items.Clear();
    for (int b=0; b<10; b++)
        for (int k=0; k<20; k++)
            for (int t = 0; t < 200; t++)
            {
                if ((20 * b + 10 * k + t == S) && (b + k + t == N))
                {
                    listBox1.Items.Add("Биків " + Convert.ToString(b));
                    listBox1.Items.Add("Корів " + Convert.ToString(k));
                    listBox1.Items.Add("Телят " + Convert.ToString(t));
                }
            }
}
```

4. Розв'яжіть задачу якщо у Вас є 300 карбованців та 400 карбованців

3. Контрольні запитання

1. Для чого використовується елементи PictureBox?
2. Як вибрати зображення для PictureBox?
3. Які є атрибути властивості SizeMode ?
4. Які існують події при роботі з PictureBox? Як їх налаштувати?
5. Наведіть перших 10 чисел Фібоначчі. Поясніть як вони обраховуються

Література

1. Евдокимов П. В. С# на примерах. СПб.: Наука и Техника, 2019. 320 с.
2. Маки А. Введение в .NET 4.0 и Visual Studio 2010 для профессионалов; пер. с англ. М. : ООО ИД "Вильямс", 2010. 416 с
3. С# 7.0. Справочник. Полное описание языка.: Пер. с англ. СПб.: ООО "Альфа-книга", 2018. 1024 с.
4. Троелсен, Эндрю, Джепикс, Филипп. Язык программирования С# 7 и платформы .NET и .NET Core. СПб. : ООО "Диалектика", 2018. 1328 с.
5. Офіційний сайт компанії Microsoft щодо технологій WPF та Win-dows Forms [Електронний ресурс]. – Режим доступу : <http://window-sclient.net>.
6. С#. Теорія та практика. URL: https://www.bestprog.net/uk/sitemap_ua/c-3
7. Сажин А. Справочник по языку программирования С#. URL: <https://brainoteka.com/blogs/c-spravochnik>.
8. С# Tutorial URL <https://www.theengineeringprojects.com>.
9. Уроки С#. URL: <https://itproger.com/course/csharp>.
10. Полное руководство по С# 8 и .NET Core. URL: <https://metanit.com/sharp/>