TEMA

Розробка застосунків на базі платформи .NET. Введення / виведення даних з використанням технології Windows Forms та мови C#



Windows Forms

Мета

Формування практичних навиків створення проєктів Windows Forms, конструювання інтерфейсу програми з допомогою візуального конструктора та елементів керування, налаштування властивостей елементів керування, використання методів і програмування обробників подій для організації введення та виведення даних засобами мови програмування С#.

Цілі заняття:

сформулювати загальний алгоритм створення візуального застосунку;

навчитись використовувати бібліотеку класів .NET для створення візуальних додатків;

спроектувати інтерфейс Windows-застосунку з використанням елементів керування;

налаштувати властивості елементів керування, застосувати методи та реалізувати обробники подій для введення / виведення даних;

перевірити коректність роботи створеного застосунку, навчитися аналізувати та усувати недоліки.

Результати навчання:

знати алгоритм створення візуальних застосунків;

вміти створювати об'єкти на основі базових класів візуальних компонентів;

аналізувати необхідність використання тих чи інших компонентів для створення візуального застосунку;

вміти налаштовувати властивості візуальних компонентів програми, використовувати методи для опрацювання даних програми;

вміти створювати обробники подій візуальних компонентів для обчислень та введення / виведення даних;

вміти виконувати тестування програми;

аналізувати помилки та знаходити оптимальні методи їх вирішення.

Що потрібно знати / вміти, щоб досягнути цілей заняття?



Цо потрібно знати / вміти для створення Windows-застосунку?

Потрібно знати синтаксис мови програмування С# / основи ООП.	?
Потрібно вміти створювати проект Windows Forms в середовищі програмування.	?
Потрібно знати бібліотеку класів для створення візуальних додатків.	?
Потрібно вміти створювати інтерфейс програми з допомогою елементів керування.	?
Потрібно вміти налаштовувати властивості елементів керування.	?
Потрібно вміти вводити / виводити дані.	?

Алгоритм створення візуального додатку

✓ Встановіть правильний порядок етапів:



Створення нового проекту	
WINDOWS FORMS.	
ЗБЕРІГАННЯ ПРОЄКТУ В ОКРЕМОМУ	
КАТАЛОЗІ.	
Конструювання інтерфейсу	
ДОДАТКУ З ДОПОМОГОЮ ЕЛЕМЕНТІВ	
КЕРУВАННЯ.	
Налаштування властивостей	
ВІЗУАЛЬНИХ КОМПОНЕНТІВ ДОДАТКУ.	
Створення обробників подій,	
НАПИСАННЯ ПРОГРАМНОГО КОДУ.	
ЗБЕРІГАННЯ ВНЕСЕНИХ ЗМІН ДО	
ПРОЄКТУ.	

Проаналізуйте, з яких елементів складаються вікна даних Windows-застосунків?



Шрифт		×
<u>Ш</u> рифт:	Ст <u>и</u> ль шрифту:	<u>Р</u> озмір:
Consolas	звичайний	11
Consolas Constantia Cooper COPPERPLATE GOTHIC Corbel CountryBlueprint	звичайний курсив жирний жирний курсив	11 ^ 12 14 16 18 20
Courier V	×	22 👻
	Зразок АаВbYyZ	Z
	Сце <u>н</u> арій:	
	Західна	~
Відобразити більше шрифті	<u>B</u>	
	ОК	Скасувати



Призначення елементів керування

Встановіть призначення елементів керування для конструювання інтерфейсу програми:

Інтерактивна вправа 2



ЕЛЕМЕНТ УПРАВЛІННЯ	НАЗВА ТА ПРИЗНАЧЕННЯ ЕЛЕМЕНТА УПРАВЛІННЯ
-	CompoRox – закритий список, дозволяє вибирате одне із значень в списку
CheddBox1	Снесквох— прапорець, застосовується для встановлення одного чи декількох параметрів
CommandButton 1	СоммалоВитто - кнопка, дозволяє ініціювати певну подію
UserForm1	UserForm – форма, контейнер для інших ofektis, має властивості вікна <u>windows</u> - додатку
Frame1	Екамк, – рамка, використовується для групування <u>обектів</u> на формі.
Label 1	LABEL – напис, призначений для відображення тексту, який користувач не може змінити з клавіатури
OptionButton1 OptionButton2	Ортонвитом - перемикач, застосовується для встановлення одного з декількох параметрів
	Техтвох – текстове поле, елемент керування, призначений для введення даних

Пригадуємо структуру ієрархії класів .Net, яка використовується у застосунках WindowsForms



Всі елементи користувацького інтерфейсу у візуальному програмуванні – це об'єкти, тобто програмні елементи, котрі мають своє відображення на екрані, містять деякі змінні, що визначають їхні властивості, деякі методи для керування об'єктами і події, на які вони можуть реагувати.

Пригадуємо щойно вивчене!





Пригадуємо спільні риси об'єктів на основі класу CONTROL

- Базовим класом для елементів керування є клас Control компонента з візуальним представленням.
- ✓ Пригадайте загальні риси для всіх класів, що відносяться до елементів графічного інтерфейсу
 - Зовнішній вигляд
 - Font, Text
 - BackColor, ForeColor
 - BackgroundImage,
 BackgroundImageLayout

- Взаємодія
 - з користувачем (перехоплення введення з клавіатури і мишки)
 - з операційною системою, та іншими програмами

Елементи даного класу мають два режими:

- Design mode використовується для візуального проектування інтерфейсу на стадії розробки, може:
 - Розміщуватись на формі
 - Задавати значення властивостей в системі програмування
- Run-Time mode використовується при роботі програми
 Змінює вигляд в залежності від дій користувача

Властивості

Властивості – певні параметри об'єкта, які визначають його стан, зовнішній вигляд тощо. Наприклад: колір, видимість, розмір, тощо.

Як задати властивість об'єкту?

1) Програмно

Необхідно ввести ім'я об'єкта, поставити крапку і за нею — ім'я властивості. Дальше ставиться знак дорівнює і значення властивості.

• Наприклад:

Form1.Opacity = 90;

2) 3 допомогою вікна властивостей

Form4 System.Windows.Forms.Fo

9 9 J 5 4 1

100		
	KeyPreview	False
	Language	(Default)
	Localizable	False
Ŧ	Location	0; 0
	Locked	False
	MainMenuStrip	(none)
	MaximizeBox	True
Ð	MaximumSize	0; 0
	MinimizeBox	True
Ŧ	MinimumSize	0; 0
	Opacity	100%
Ŧ	Padding	0; 0; 0; 0
	RightToLeft	No
	RightToLeftLayou	False
	ShowIcon	True
	ShowInTaskbar	True

Методи та події

• Методи, тобто команди, які можна застосувати до об'єкта.

Наприклад, у форми є метод Close(), який закриває вікно форми

- Подія являє собою певну дію, на яку реагує об'єкт (наприклад, клацання мишею чи натискання на клавішу), для котрої можна запрограмувати відповідь.
- Події виникають в результаті дій користувача, програми, або вони можуть бути викликані системою.

```
private void button5_Click(object sender, EventArgs e)
{
   textBox1.Text = textBox1.Text + "1";
}
```

✓ Пригадайте синтаксис властивостей, методів та обробників подій на мові Visual C#:

Інтерактивна вправа 3



Form1.ForeColor = Blue; Button1.Text = "Οκ";	Властиві сть
<pre>Application.Run(Form1); Application.Exit();</pre>	Метод
<pre>private void button1_Click (object sender, EventArgs e) { string s = textBox1.Text; }</pre>	Обробник події

✓ Пригадайте синтаксис властивостей, методів та обробників подій на мові Visual C#:

ToString()

b.Top = 50; b.Left = 60;

int lenght = textBox1.Text.Length - 1;

textBox1.Clear();

float.Parse()

private void button5_Click(object sender, EventArgs e)
{textBox1.Text = textBox1.Text + "1"; }



Ô

3 використанням вивчених інструментів **створити «калькулятор» для додавання,** *віднімання, множення* та *ділення* **довільних чисел**

Практичне завдання

Етапи виконання завдання

✓ Аналізуємо етапи виконання завдання з допомогою карти знань



1. Проєктування інтерфейсу, налаштування властивостей елементів керування

- Які елементи керування потрібно використати, щоб користувач мав можливість ввести числа, виконати обчислення та побачити результат?
- ✓ Насправді варіантів проектування інтерфейсу такого додатку може бути декілька!



Спробуємо реалізувати даний варіант інтерфейсу

3 допомогою візуального конструктора

Програмно

Алгоритм роботи такий:

- 1) Обираємо потрібний елемент на панелі елементів керування
- 2) поміщаємо його на форму
- 3) налаштовуємо необхідні властивості у вікні властивостей



Алгоритм роботи такий:

1) Пишемо програмний код для створення нових елементів на основі базових класів (у функції Form_Load() — завантаження форми, яка виконується першою при запуску проєкту),

//1) створюємо кнопку
Button b = new Button();

2) задаємо властивості елементам керування //2) задаємо властивості //позиція на екрані b.Top = 50; b.Left = 60; //текст кнопки b.Text = «1"; //колір кнопки b.BackColor = Color.FromArgb(200, 12, 15); //розмір b.Size = new System.Drawing.Size(45, 45); // шрифт -> гарнітура - розмір - накреслення b.Font = new Font("Arial", 14, FontStyle.Regular);

3) додаємо ці елементи в колекцію елементів керування форми

```
//3) додаємо на форму кнопку
this.Controls.Add(b);
```



3. Тестування

Перевіряємо правильність обчислень

- 🗸 При введенні цілих чисел
- При введенні дробових чисел
- При виборі арифметичної операції

Які типи помилок бувають?

- Синтаксичні
- Логічні

 ERORR

- Неправильні результати обчислень
 Не виводяться дані у потрібному форматі
- Аварійне завершення роботи програми через неправильні дії користувача!
- 1 провина програміста

 \succ

 \geq

 \succ

- 2 провина програміста
- 3? також провина програміста! Чому?



4. Усунення недоліків

• При введенні даних необхідно їх перевіряти на відповідність? • унеможливити введення некоректних даних — найкращий варіант

> Забезпечити натискання / зчитування арифметичного оператора та крапки лише один раз



Підсумки. Навчальний тест

http://moodle.gi.edu.ua/mod/quiz/view.php?id=46145

Питання 1 Відповіді ще не було Макс. оцінка до 1,00 у∾ Відмітити питання	Створювати обробники подій можна в режимі візуального застосунку	Перехід По Тесту 1 2 3 4 5 6 7 8 9 10 Завершити спробу
питання Федагувати питання		РОЗПОЧАТИ НОВУ СПРОБУ



Підсумки

• Що нового ви навчилися на занятті?



Цілі заняття:

сформулювати загальний алгоритм створення візуального застосунку;

навчитись використовувати бібліотеку класів .NET для створення візуальних додатків;

спроектувати інтерфейс Windows-застосунку з використанням елементів керування;

налаштувати властивості елементів керування, застосувати методи та реалізувати обробники подій для введення / виведення даних;

перевірити коректність роботи створеного застосунку, навчитися аналізувати та усувати недоліки.

Результати навчання:

знати алгоритм створення візуальних застосунків;

вміти створювати об'єкти на основі базових класів візуальних компонентів;

аналізувати необхідність використання тих чи інших компонентів для створення візуального застосунку;

вміти налаштовувати властивості візуальних компонентів програми, використовувати методи для опрацювання даних програми;

вміти створювати обробники подій візуальних компонентів для обчислень та введення / виведення даних;

вміти виконувати тестування програми;

аналізувати помилки та знаходити оптимальні методи їх вирішення.

Самостійна робота. Додавання функціоналу

✓ Видаляємо останній введений символ (кнопка <-)</p>

```
private void button2_Click(object sender, EventArgs e)
{
    int lenght = textBox1.Text.Length - 1;
    string text = textBox1.Text;
    textBox1.Clear();
    for (int i = 0; i < lenght; i++)
    {
        textBox1.Text = textBox1.Text + text[i];
    }
</pre>
```



Самостійна робота. Додавання функціоналу

✓ ЗМІНЮЄМО ЗНАК ВВЕДЕНОГО ЧИСЛА

```
private void button1_Click(object sender, EventArgs e)
{
    if(znak==true)
    {
        textBox1.Text = "-" + textBox1.Text;
        znak = false;
    }
    else if (znak==false)
    {
        textBox1.Text=textBox1.Text.Replace("-", "");
        znak = true;
    }
```



Домашне завдання

- Додати можливість виконання інших арифметичних операцій
- Додати можливість застосування математичних функцій
- Оформити звіт по виконаній роботі.
- Повторити теоретичні відомості про організацію циклічних обчислень, синтаксис операторів циклу, базові алгоритми опрацювання масивів.