

Функція обробки події.

Подієво-орієнтована модель програмування.

Програми з графічним інтерфейсом працюють за подієво-орієнтованим принципом. Це означає, що виконання програми визначається подіями, які виникають у процесі взаємодії користувача з інтерфейсом.

Типові події:

- натискання кнопки;
- рух або клацання миші;
- введення тексту з клавіатури;
- завантаження або закриття форми;
- зміна значення елемента керування.

Поняття події у Windows Forms.

Подія — це сигнал, який повідомляє програму про те, що відбулася певна дія.

У WinForms:

- кожен елемент керування (Button, TextBox, Form тощо) має набір подій;
- подія пов'язується з функцією-обробником (event handler);
- при виникненні події автоматично викликається відповідна функція.

Функція обробки події (Event Handler).

Функція обробки події — це спеціальна функція, яка виконується у відповідь на виникнення події.

Загальні характеристики:

- має визначений сигнатурою формат;
- прив'язується до конкретної події конкретного елемента;
- містить код, що виконує необхідні дії

Основні характеристики Windows Forms:

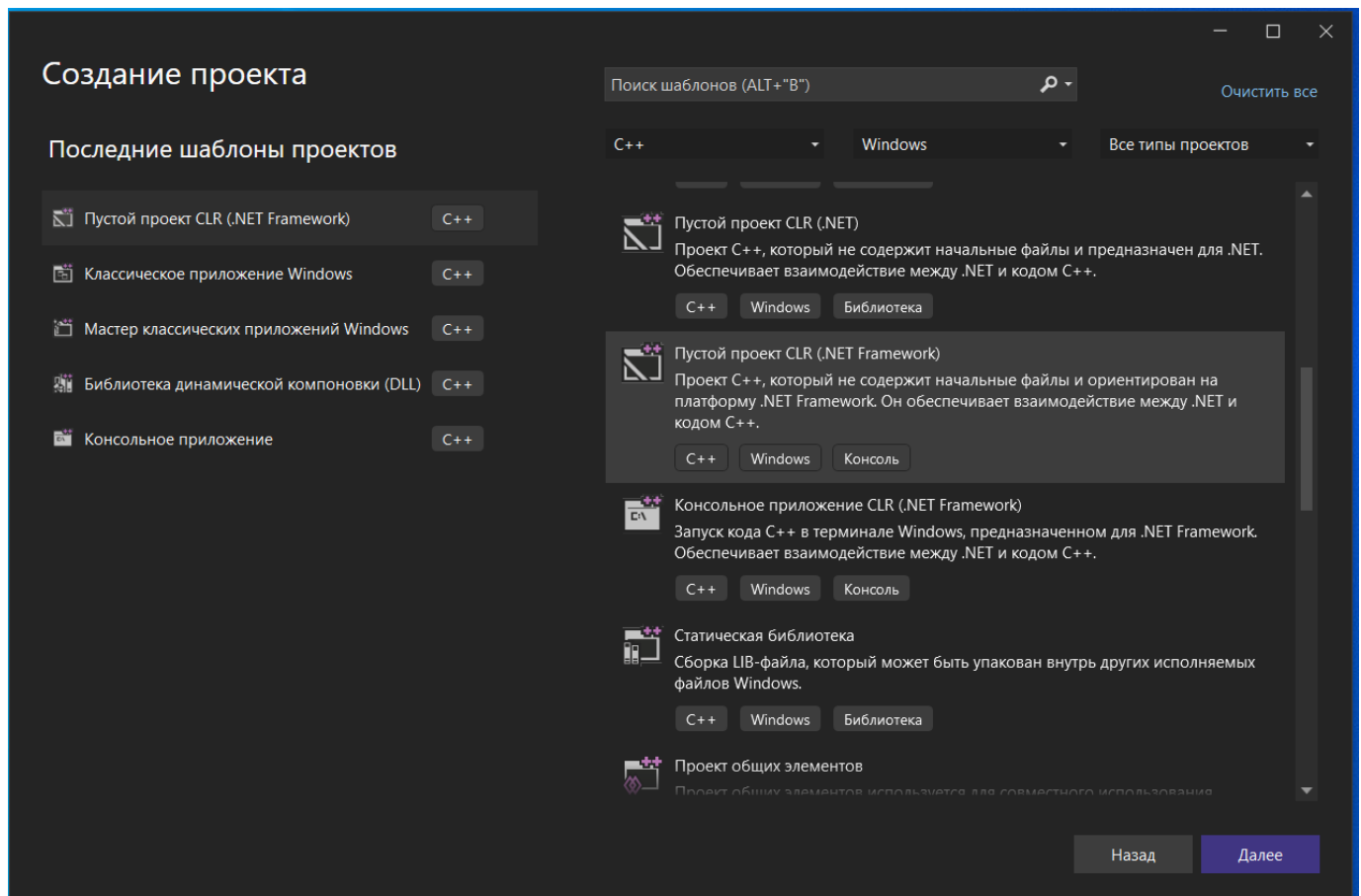
1. **Бібліотека компонентів:** Windows Forms надає широкий набір готових до використання компонентів, таких як текстові поля, кнопки, лістбокси, табличні дисплеї тощо, які можна легко перетягувати на форму в середовищі розробки, такому як Visual Studio.
2. **Подієво-орієнтоване програмування:** Інтерфейси користувача, створені за допомогою Windows Forms, базуються на моделі обробки подій. Це означає, що програма реагує на дії користувача, такі як кліки мишею, натискання клавіш тощо, за допомогою подій, до яких можна прив'язати обробники.
3. **Інтеграція з .NET Framework:** Програми на основі Windows Forms легко інтегруються з іншими компонентами .NET Framework, такими як ADO.NET для доступу до даних, серіалізація для збереження стану, та асинхронне програмування.
4. **Кастомізація і розширення:** Користувачі можуть не тільки використовувати стандартні компоненти, але й створювати власні, наслідуючи від базових класів, що дозволяє розширити функціональність стандартних компонентів.
5. **Візуальна розробка:** Visual Studio надає потужні засоби для візуального проектування Windows Forms додатків, включаючи дизайнер форм, який дозволяє перетягувати компоненти на форму, автоматично генеруючи відповідний код.

Приклади використання:

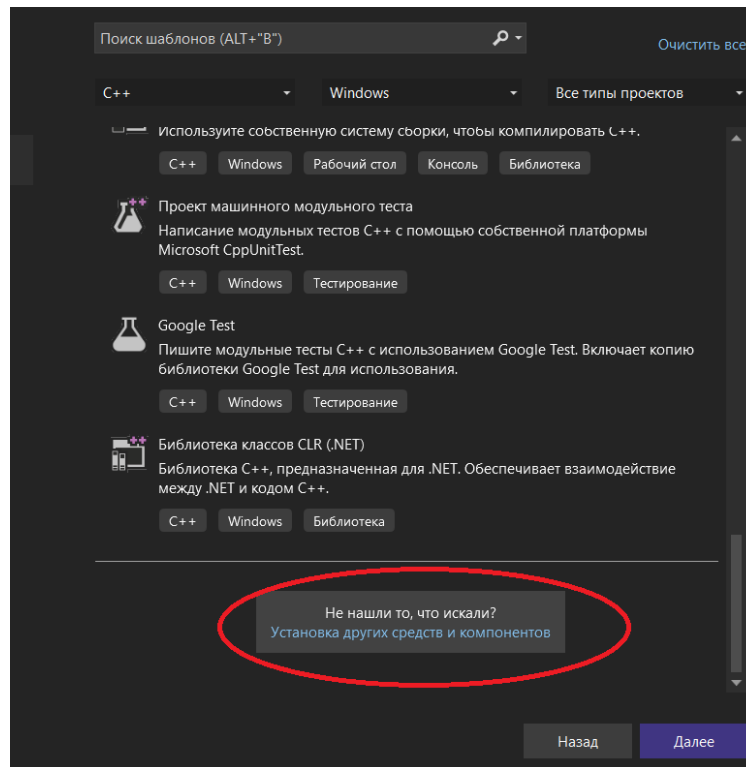
- **Бізнес-додатки:** Windows Forms часто використовується для створення бізнес-додатків, що вимагають складної логіки та інтеракцій з користувачем, а також інтеграції з базами даних.

- **Утиліти для Windows:** Малі утиліти, такі як конвертери файлів, редактори тексту та інші інструменти, часто розробляються з використанням Windows Forms через простоту створення та розповсюдження таких додатків.

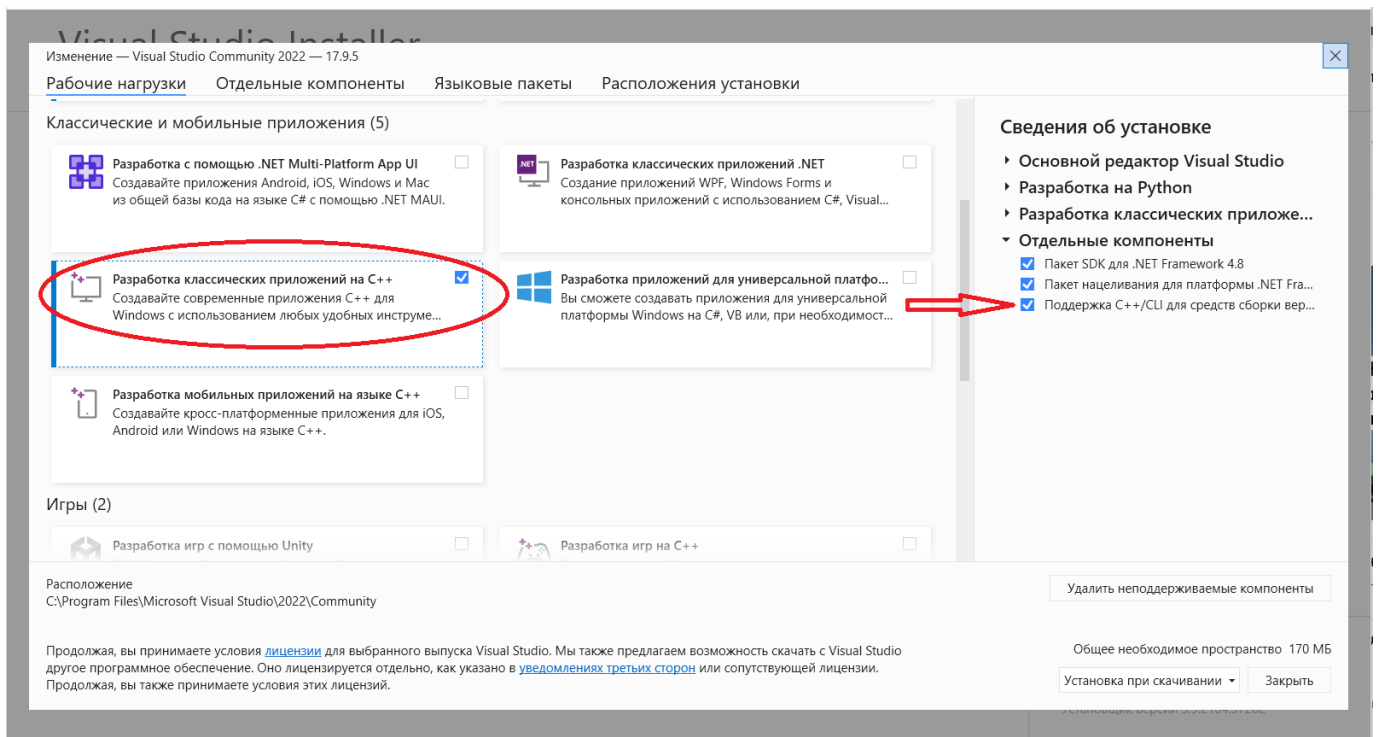
Windows Forms продовжує підтримуватися Microsoft, хоча компанія заохочує розробників розглядати сучасніші технології для нових проектів, такі як WPF (Windows Presentation Foundation) або UWP (Universal Windows Platform) для більш сучасних інтерфейсів та можливостей.



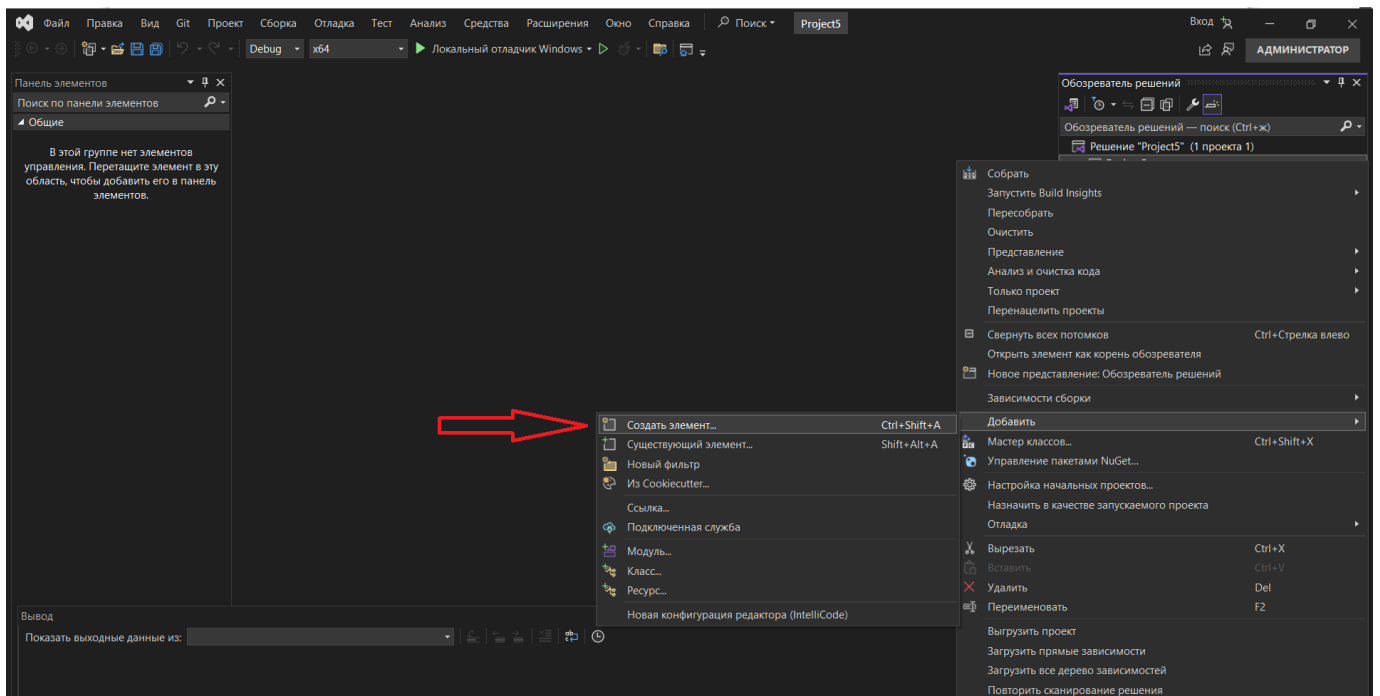
Крок 1



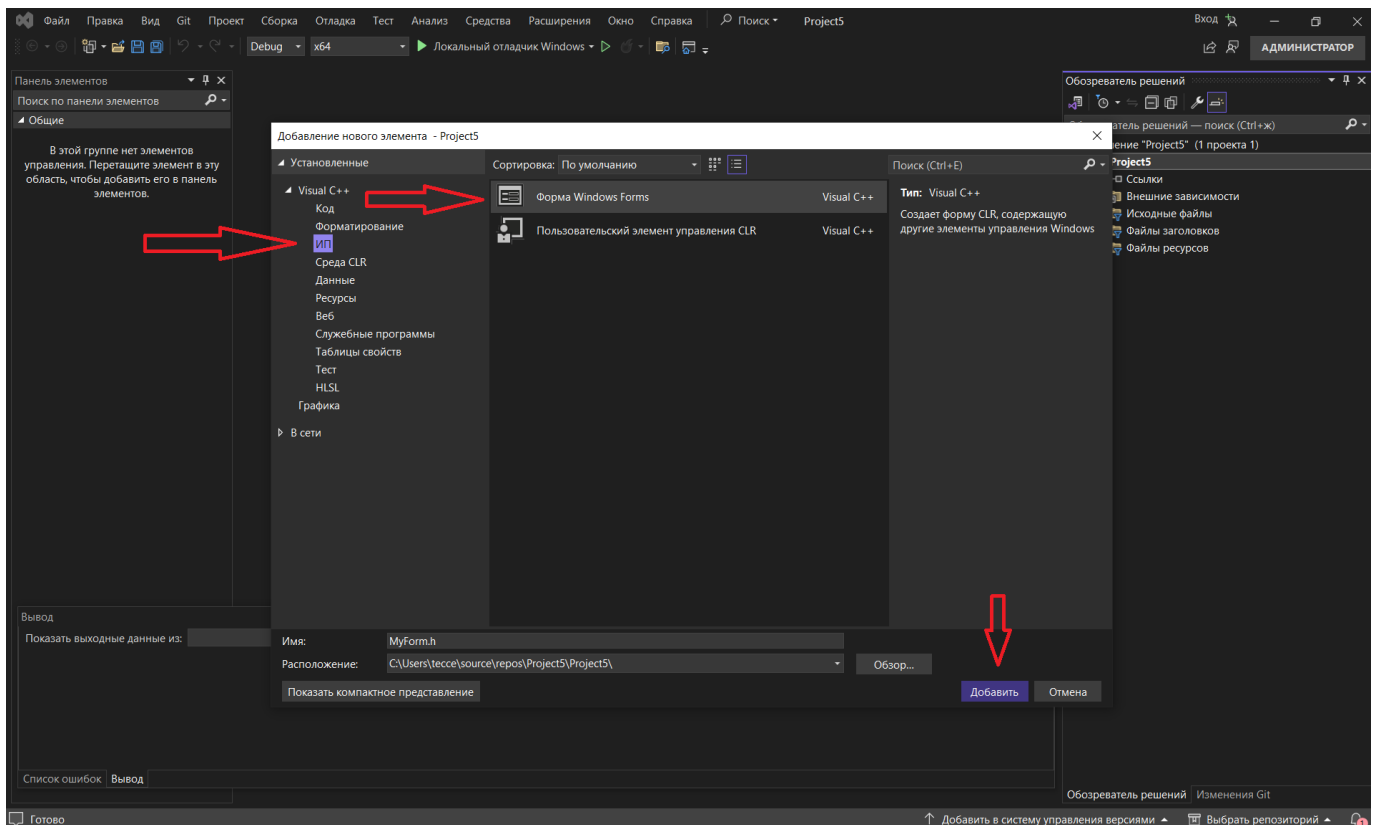
Крок 2



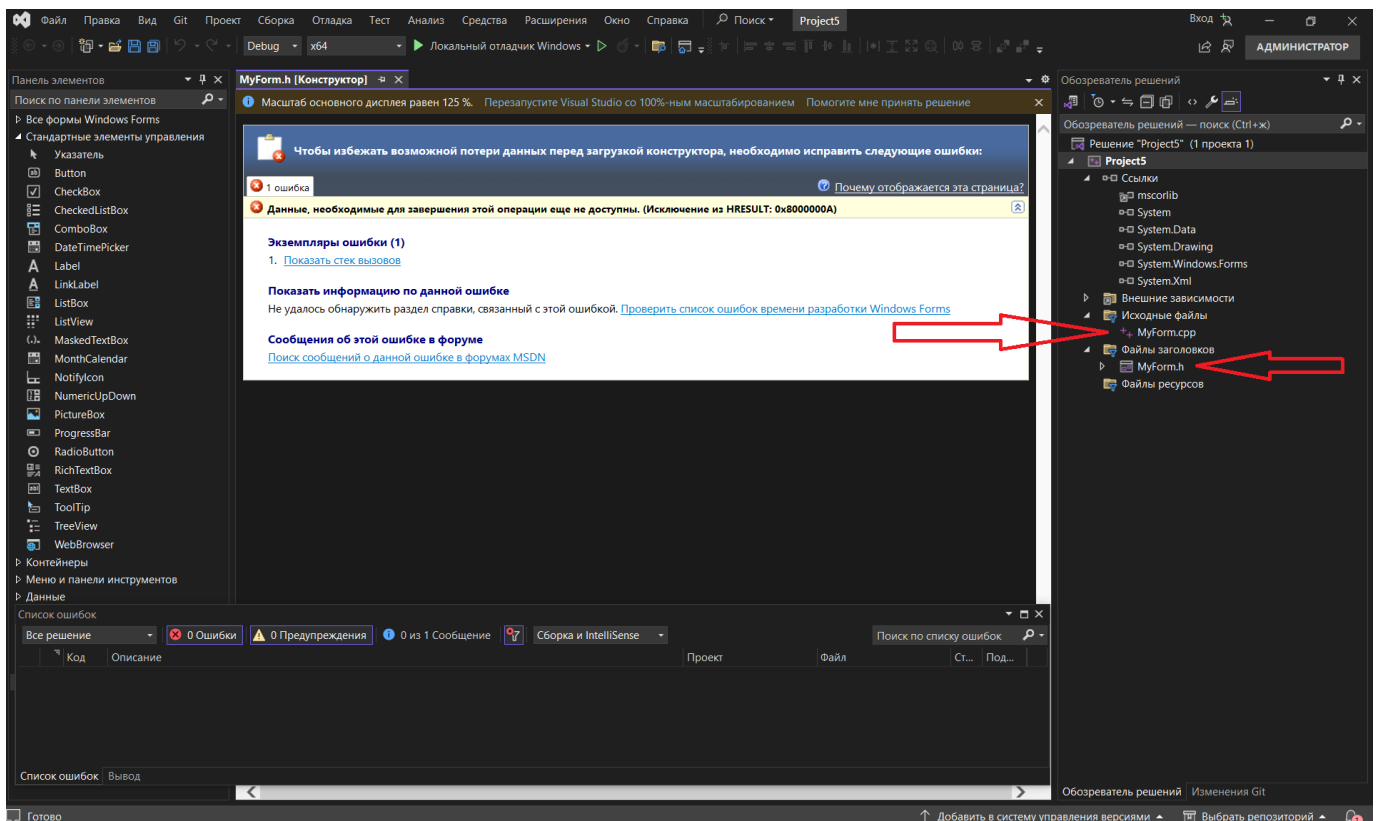
Крок 3



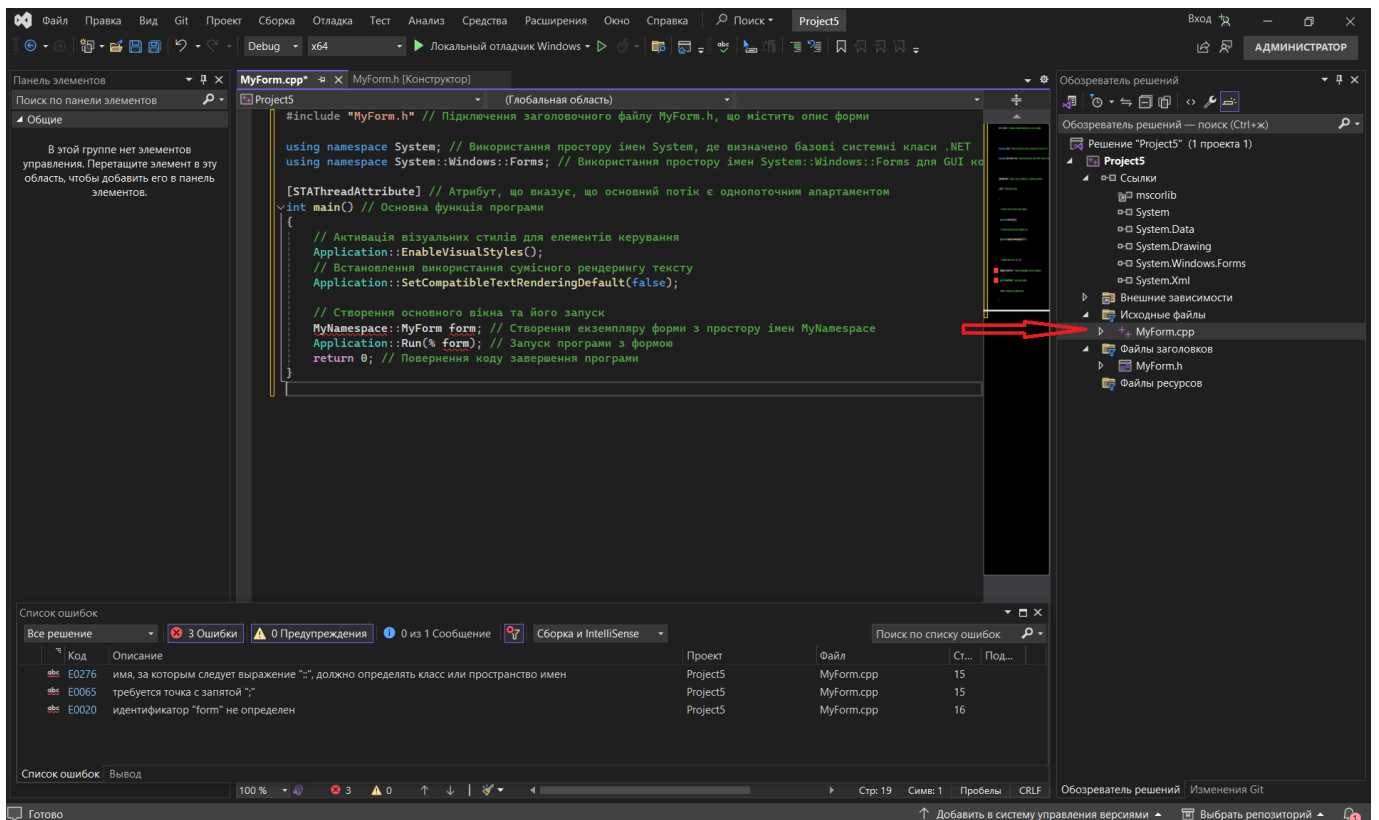
Крок 4



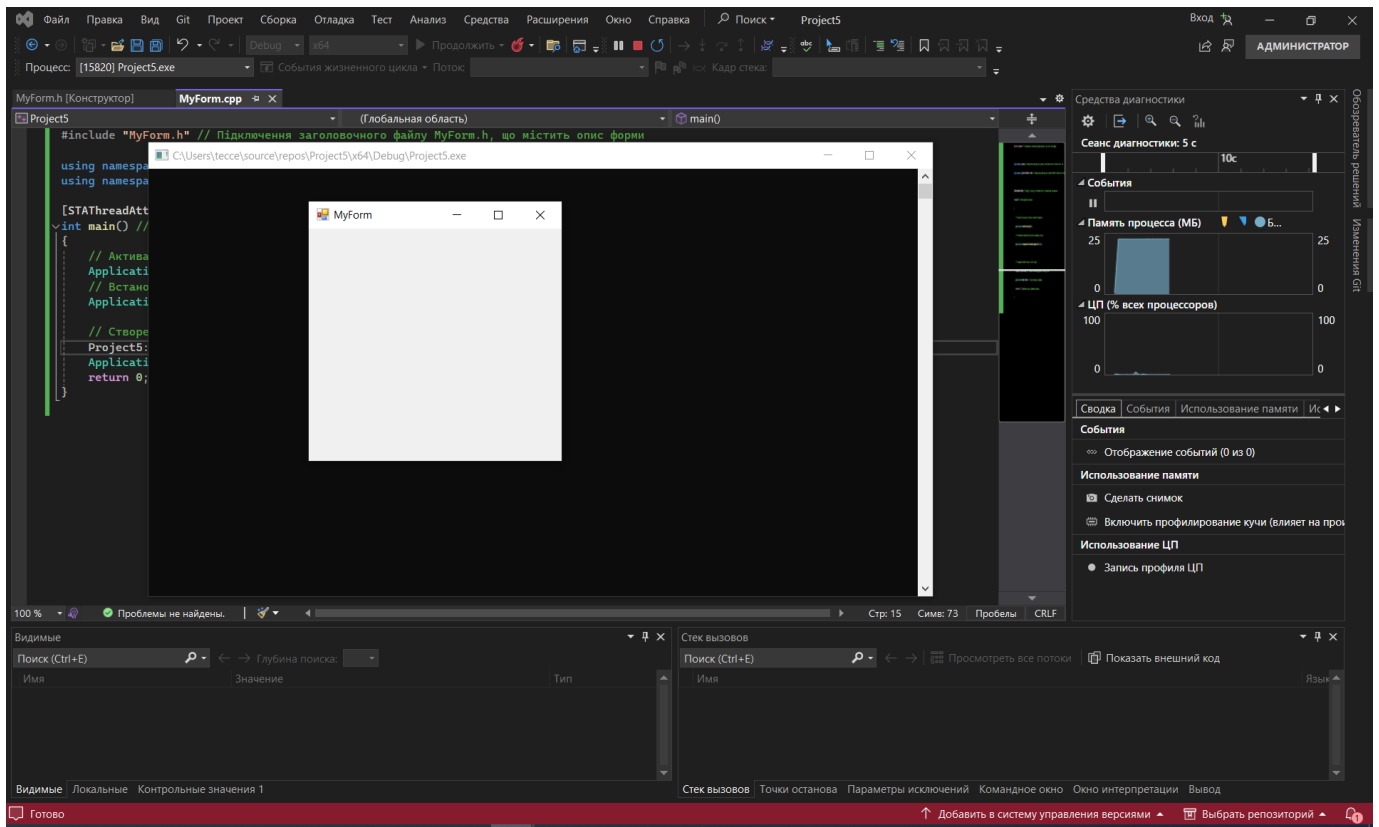
Крок 5



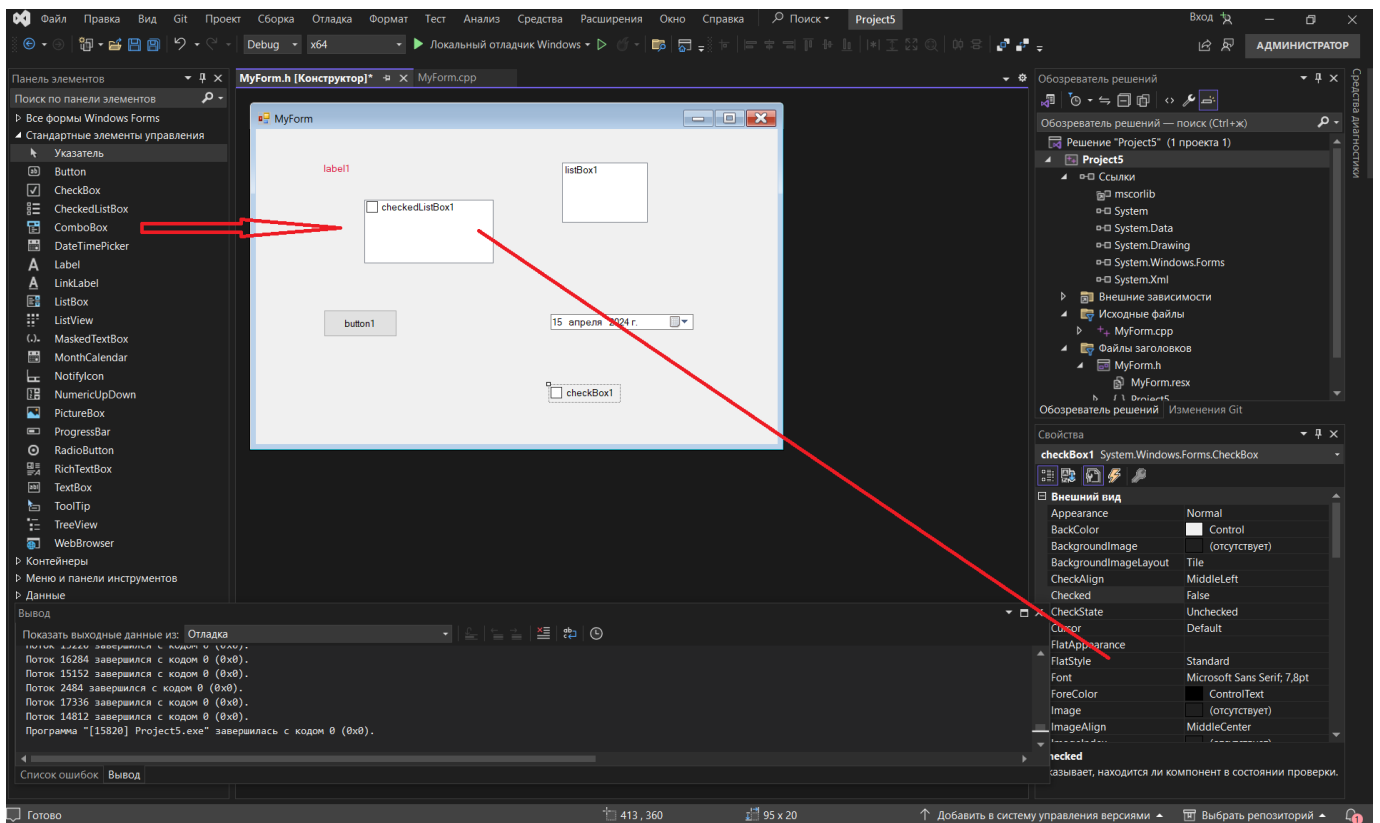
Крок 6



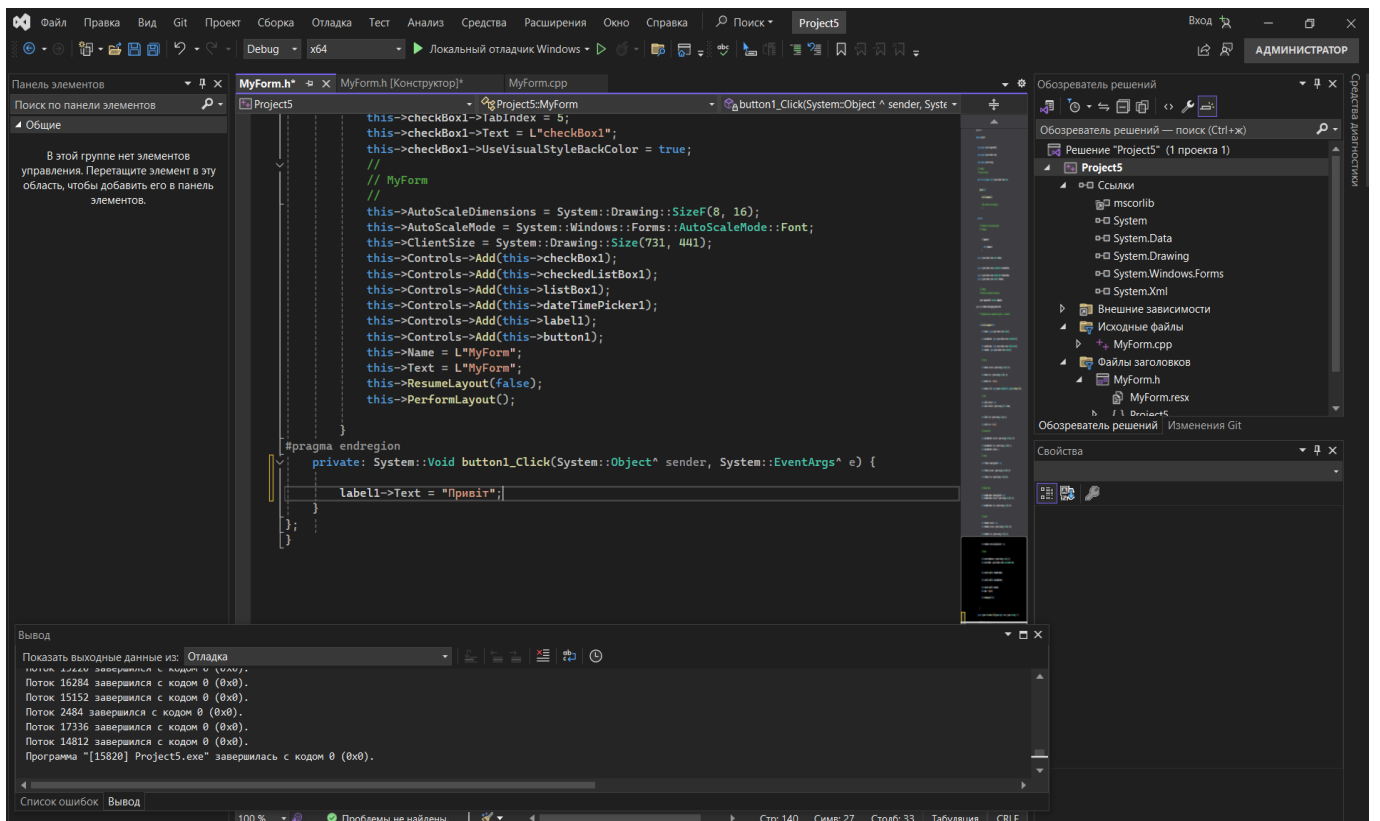
Крок 7



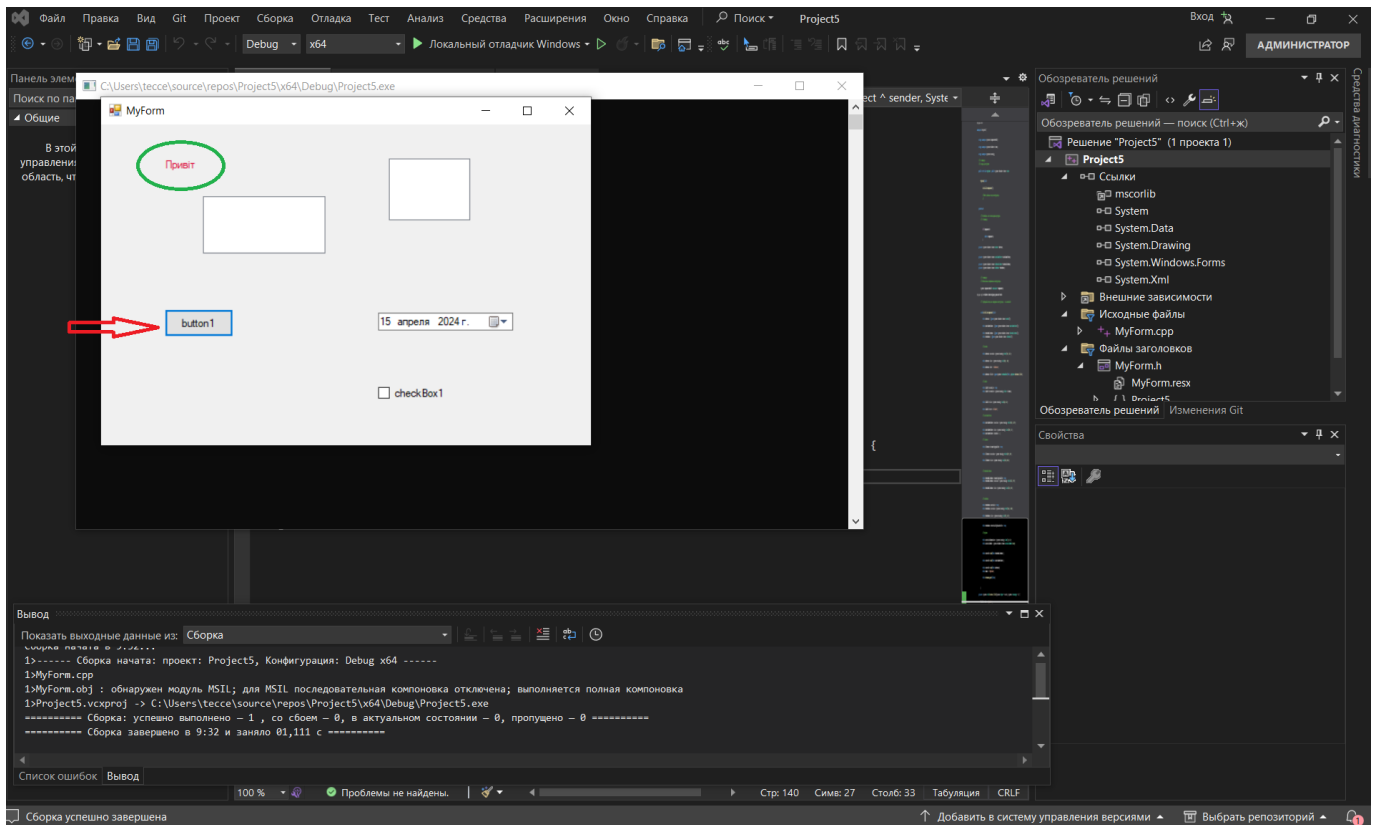
Крок 8



Крок 9



Крок 10



Крок 11

Висновки:

- Платформа Windows Forms:** Ми дізналися, що Windows Forms — це потужний інструмент для створення графічних інтерфейсів користувача в додатках Windows, який є частиною .NET Framework. Ця технологія дозволяє розробникам створювати візуально привабливі десктопні додатки з розширеними можливостями управління даними та інтерфейсом.
- C++/CLI як мова реалізації:** Ми розглянули особливості використання C++/CLI, мови, яка поєднує класичний C++ з можливостями .NET, що дозволяє з легкістю інтегрувати управління пам'яттю і виклики .NET-бібліотек.

3. **Структура додатку на Windows Forms:** Ми проаналізували типову структуру програми Windows Forms, включаючи основні компоненти, такі як форми та компоненти, а також основи подієво-орієнтованого програмування для реагування на дії користувачів.
4. **Візуальна розробка:** Було підкреслено важливість візуального дизайнера в Visual Studio, який спрощує процес розробки інтерфейсів, дозволяючи драг-енд-дроп методом розташовувати компоненти на формах.
5. **Практичне застосування:** Ми вивчили простий приклад коду, який демонструє створення і запуск форми у Windows Forms за допомогою C++/CLI, розглянувши ключові методи та їх використання.

Незважаючи на наявність новіших технологій для розробки інтерфейсів, Windows Forms залишається важливим інструментом для створення традиційних десктопних додатків у середовищі Windows, особливо коли йдеться про проєкти, які потребують швидкого створення.