Міністерство освіти України

Національний технічний університет "ХПІ"

кафедра “ННІКНІТ"

**Звіт**

**Лабораторна робота 2**

з дисципліни "Програмна технологія .NET"

Виконав: студент групи КН-921Г

Верещак А. Г.

Перевірив:

Івашко А.В.

Харків 2023

Зміст

[Завдання 3](#__RefHeading___Toc925_3658079158)

[Використання конструктора форм для створення та налаштування інтерфейсу віконних застосунків в мові C# 4](#__RefHeading___Toc927_3658079158)

[Хід роботи 5](#__RefHeading___Toc929_3658079158)

[Завдання 1 та 2: 5](#__RefHeading___Toc931_3658079158)

[Завдання 3, 4: 6](#__RefHeading___Toc2241_3286642050)

[Завдання 5, 6: 7](#__RefHeading___Toc2243_3286642050)

[Завдання 7, 8: 8](#__RefHeading___Toc2245_3286642050)

[Висновок 9](#__RefHeading___Toc933_3658079158)

Лабораторна робота №3. **Використання конструктора форм для створення та налаштування інтерфейсу віконних застосунків. Частина 1**

**Мета роботи**: вивчення властивостей та використання компонентів конструктора форм для створення і налаштування віконного інтерфейсу застосунків

# Завдання

* 1. Використовуючи компонент Label та його відповідні властивості, створити застосунок, вікно якого містить три рядки будь-якого тексту, які відрізняються один від одного аз розміром, кольором, типом вирівнювання та формою тексту згідно із параметрами свого варіанту
  2. Виконати минуле завдання додавши рядки тексту програмним шляхом, тобто під час завантаження вікна застосунку
  3. Виконати перше завдання, використавши замість компоненту Label компонент TextBox
  4. Виконати минуле завдання додавши текстові поля програмним шляхом, тобто під час завантаження вікна застосунку
  5. Використовуючи компонент PictureBox та його відповідні властивості створити форму, що містить групу будь-яких зображень за топологією згідно із параметрами свого варіанту (враховувати приблизні співвідношення, а не абсолютні розміри)
  6. Виконати минуле завдання додавши зображення програмним шляхом, тобто під час завантаження вікна застосунку
  7. Використовуючи компоненти Button, TableLayoutPanel та відповідні властивості створити форму, що містить групу кнопок за топологією згідно із параметрами свого варіанту
  8. Виконати минуле завдання додавши кнопки та таблицю розмішення програмним шляхом, тобто під час завантаження вікна застосунку

# Використання конструктора форм для створення та налаштування інтерфейсу віконних застосунків в мові C#

*Це процес створення користувацького інтерфейсу за допомогою графічного інтерфейсу (GUI) у середовищі розробки, такому як Visual Studio. У C# інтерфейси користувача часто створюються з використанням Windows Forms (WinForms) або WPF (Windows Presentation Foundation). Давайте розглянемо основні кроки створення та налаштування інтерфейсу з використанням WinForms.*

1. **Відкрийте середовище розробки:** Відкрийте Visual Studio або іншу IDE для роботи з C#. Створіть новий проект Windows Forms Application.
2. **Додайте форму:** В проекті за замовчуванням зазвичай є одна форма (Form). Ви можете додавати додаткові форми, які будуть використовуватися для різних вікон та інтерфейсних частин вашого додатку.
3. **Додайте елементи керування:** До форми можна додавати різні елементи керування, такі як кнопки (Button), текстові поля (TextBox), мітки (Label), списки (ListBox або ComboBox) і так далі. Ці елементи використовуються для взаємодії з користувачем.
4. **Налаштуйте властивості:** Властивості кожного елемента керування можна налаштовувати властивостями властивостей. Наприклад, ви можете змінити розмір, колір фону, шрифт тексту тощо.
5. **Розміщення елементів:** Для розміщення елементів на формі можна використовувати різні контейнери, такі як панелі (Panel), таблиці (TableLayoutPanel), групові контейнери (GroupBox) і т. д. Це допомагає організувати інтерфейс і забезпечити його правильне розташування на формі.
6. **Обробка подій:** Ви також можете додавати обробники подій до елементів керування, які відповідають за реакцію програми на взаємодію користувача. Наприклад, ви можете додати обробник події "клік" до кнопки, щоб виконувати певні дії при її натисканні.
7. **Програмуйте логіку:** Після створення і налаштування інтерфейсу ви можете програмувати логіку застосунку в C#. Це включає в себе обробку подій, зчитування та запис даних, взаємодію з базами даних, обробку помилок і багато іншого.
8. **Перевірка та налагодження:** Перед випуском програми перевірте та налагодьте її, щоб переконатися, що вона працює правильно і задовольняє всі ваші вимоги.
9. **Публікація:** Після успішного створення програми і її налагодження ви можете опублікувати її для використання користувачами.

*Конструктор форм у сполученні з мовою програмування C# дозволяє швидко та ефективно створювати інтерфейси для віконних застосунків, спрощуючи роботу з графічними елементами та їх властивостями.*

# Хід роботи

# Завдання 1 та 2:

Створюємо новий проект Task\_1, Task\_2.

Додаємо до форми компоненти Label, та модифікуємо за завданням:

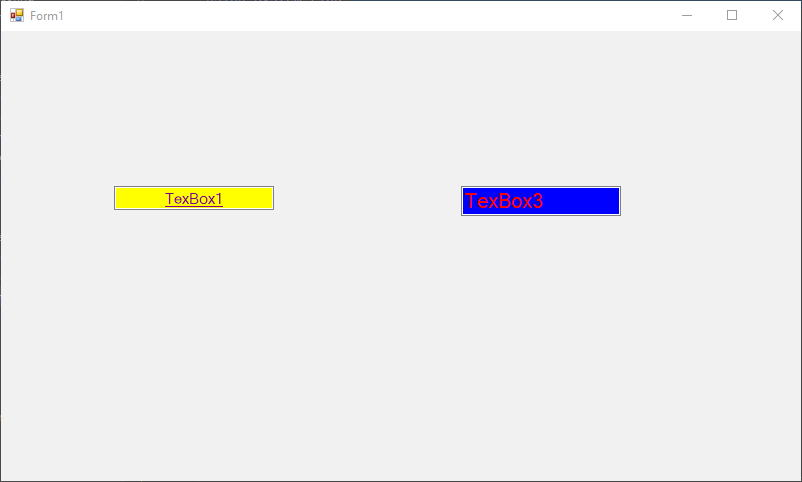


Рисунок 5.1 — Форма Task\_1 та Task\_2

# Завдання 3, 4:

Створюємо новий проект Task\_3, Task\_4

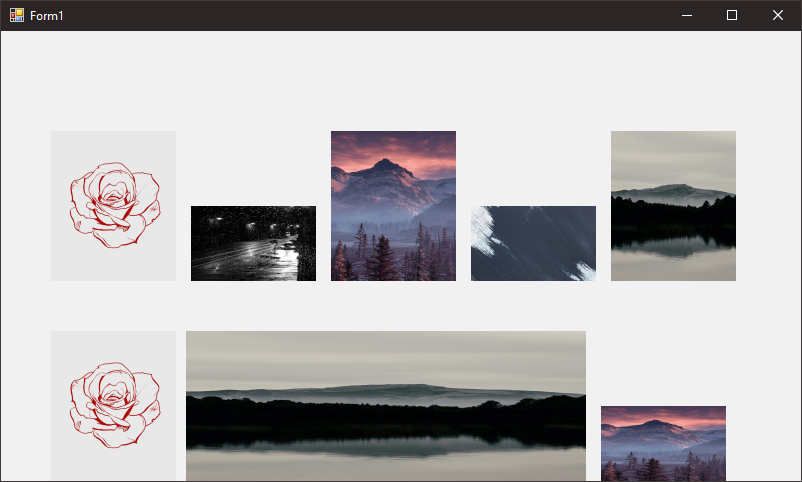
Додаємо до форми компоненти TextBox, та модифікуємо за завданням:

Рисунок 5.2 — Форма Task\_3 та Task\_4

# Завдання 5, 6:

Створюємо новий проект Task\_5, Task\_6.

Додаємо до форми компоненти PictureBox, та модифікуємо за завданням:

Рисунок 5.3 — Форма Task\_5 та Task\_6

# Завдання 7, 8:

Створюємо новий проект Task\_4.

Додаємо до форми компоненти Button, та розставляємо за завданням:

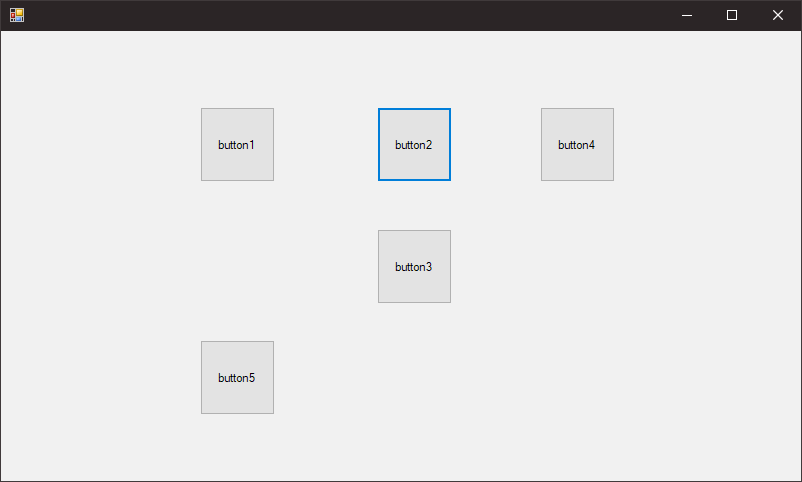


Рисунок 5.4 — Форма Task\_7 та Task\_8

# Висновок

Використання конструктора форм для створення та налаштування інтерфейсу віконних застосунків є надзвичайно важливим і потужним інструментом для розробки програмного забезпечення. В даному висновку ми підсумуємо ключові аспекти цієї теми та її важливість.

Конструктори форм, доступні в середовищах розробки, таких як Visual Studio, дозволяють розробникам створювати інтерфейси користувача швидко та ефективно. Ця зручність особливо важлива в роботі над віконними додатками, де інтерфейс користувача відіграє ключову роль у сприйнятті програми.

У створенні інтерфейсу за допомогою конструктора форм важливо враховувати кілька ключових аспектів:

1. Доступність і простота: Конструктори форм роблять створення інтерфейсу доступним навіть для розробників без великого досвіду. Це дозволяє більшій кількості людей створювати візуально привабливі та функціональні програми.

2. Гнучкість і керованість: Конструктори форм надають можливість точно налаштовувати кожен елемент інтерфейсу, встановлювати його властивості, розміщення та поведінку. Це дає розробникам повний контроль над тим, як їх програми виглядають і функціонують.

3. Швидкість розробки: Використання конструктора форм значно прискорює процес розробки, оскільки розробники можуть створювати інтерфейси шляхом перетягування та налаштування елементів, а не писати код вручну.

4. Зручність тестування і налагодження: Інтерфейс, створений за допомогою конструктора форм, легше тестувати та налагоджувати, що дозволяє виявляти та виправляти помилки швидше та ефективніше.

5. Підтримка відомих платформ: Більшість конструкторів форм підтримують популярні платформи розробки, такі як C# з використанням Windows Forms або WPF, що робить їх ідеальними для розробки для операційних систем Windows.

Зважаючи на ці фактори, використання конструктора форм для створення і налаштування інтерфейсу віконних застосунків є критично важливим етапом у розробці програмного забезпечення. Він дозволяє розробникам створювати додатки, які не лише відповідають потребам користувачів, але і виглядають привабливо та працюють ефективно. Це спрощує процес розробки і сприяє популярності програм, що, в свою чергу, може призвести до більшого успіху розробників і задоволеності користувачів.