Змістовий модуль 1
Організація процедурно - орієнтованих програм
Лабораторна робота 1
Інтегроване середовище системи програмування Visual Studio .NET

Мета роботи – ознайомлення з інтерфейсом, основними поняттями та можливостями системи програмування Visual C# .NET на прикладі виконання найпростіших програм.

Ця лабораторна робота має демонстраційний характер, тобто індивідуальне завдання відсутнє. Відповідно до Національної рамки кваліфікацій вона сприяє напрацюванню таких **компетентностей**:

*знання***:**

основних компонентів інтерфейсу системи програмування Visual C# .NET;

порядку підготовки C# програми для подальшого виконання;

структури типової програми мовою C# у консольному та графічному виконаннях;

*уміння***:**

налаштовувати систему програмування з урахуванням заданих вимог;

працювати з редактором тексту в середовищі Visual C# .NET;

виконувати компіляцію, налагодження та запуск готових демонстраційних програм;

отримати роздруківку вихідного тексту програми і результату її роботи;

*комунікації***:**

аргументована взаємодія з клієнтами та замовниками під час вибору середовища розроблення та виконання програмних продуктів;

рекомендації команді учасників проекту щодо налаштовування системи програмування Visual C# .NET;

автономність і відповідальність:

самостійне формулювання рекомендацій щодо оптимізації процесу підготовки програм до виконання;

прогнозування вигляду результатів виконання консольних і графічних програм у середовищі Visual C# .NET.

Основні положення

Базова технологія безпосередньо пов'язана з мовою C#, має назву .NET (вимовляється як "дот нет").

.NET – це загальний термін для багатьох служб, які надаються й використовуються під час створення та виконання програми на C#.

Мови програмування та компілятори.

Традиційно перетворення вихідного коду, написаного мовою високого рівня, в машинний код здійснювали системні програми, які називаються *компіляторами*.

На рис. 1 наведено ілюстрацію того, як вихідний код типової мови високого рівня перетворюється у програму, що виконується.

Рис. 1. **Традиційний процес компіляції**

Написаний текст, який містить інструкції мови високого рівня, називається *вихідним кодом*.

У випадку C# цей вихідний код зберігається в файлі з розширенням .cs.

Результатом компіляції стає *програма, що виконується*, яка складається з інструкцій машинної мови.

Елементи інтерфейсу Visual Studio .NET.

Опис інтерфейсу наведено у двох варіантів: для версій Visual Studio .NET Ultimate 2012, а також для Visual Studio .NET 2019.

VS Ultimate 2012

Основні елементи інтерфейсу містять величезний набір інструментів, покликаних спростити життя програмісту. У цій роботі зроблено огляд деяких можливостей середовища Visual Studio .NET. Багато пунктів меню та керівні вікна будуть описані далі у міру виконання поточних лабораторних робіт.

Стартова сторінка

Для запуску Visual Studio .NET слід вибрати пункт меню Пуск / Програми / Microsoft Visual Studio .NET / Microsoft Visual Studio .NET.

На екрані з'явиться стартова сторінка Visual Studio Ultimate 2012, фрагмент якої зображено на рис. 2.

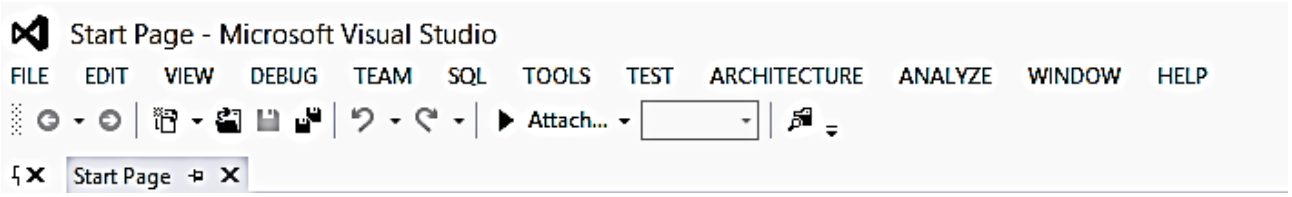


Рис. 2. **Фрагмент стартової сторінка Visual Studio .NET (Ultimate 2012)**

Visual Studio .NET — це не тільки середовище для розроблення додатків мовою C#. Воно дозволяє створювати додатки мовами VB, C#, C ++, формувати Setup (інсталяційний пакет) програм і багато іншого.

Для того, щоб реально побачити, як створюється новий проект у Visual Studio .NET, слід вибрати пункт меню Start / New / Project. Після його виклику з'явиться вікно, аналогічне зображеному на рис. 3.

Тут можна вибрати потрібну мову програмування (в лівій частині вікна) або якийсь спеціальний майстер створення додатків — цей список може поповнюватися інструментами незалежних розробників. Оскільки вивчається мова програмування C#, слід вибрати пункти Visual C# / Windows.

У правій частині вікна потрібно вказати тип створюваного проекту. Це може бути Windows-додаток (Windows Form Application), додаток для Інтернету (WPF Browser Application), консольний додаток (Console Application) і деякі інші.

Вибрати в лівій частині вікна пункт Windows Application, а в правій частині — Windows Form Application. Крім того, можна вказати назву створюваного проекту і шлях до каталогу, в якому він буде розташовуватися. Натиснути ОК.

Тепер можна побачити основні частини візуальної середовища розробки проекту. Вони зображені на рис. 4.

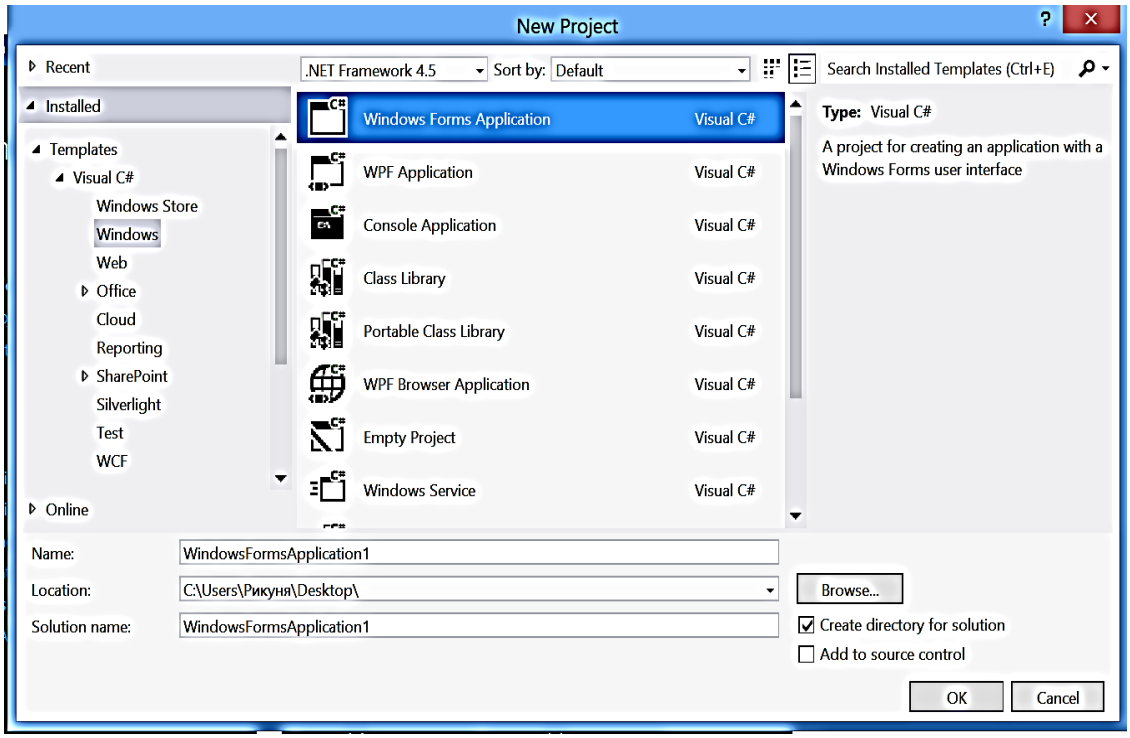


Рис. 3. **Вікно вибору типу проекту**

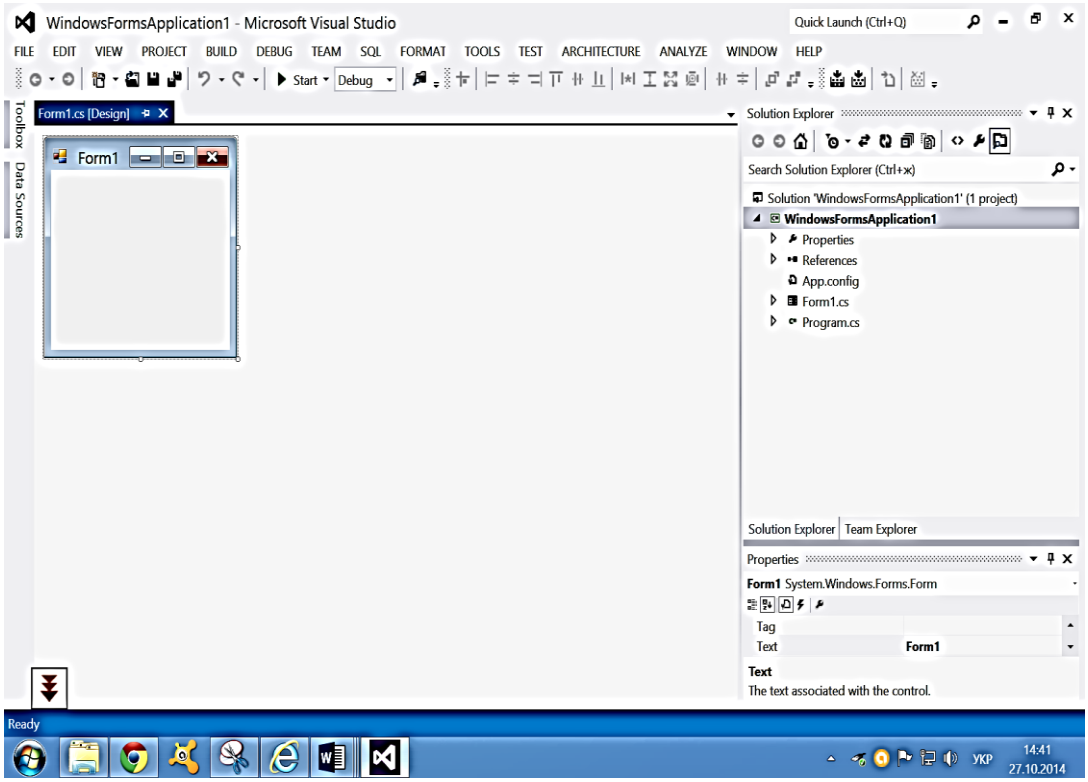
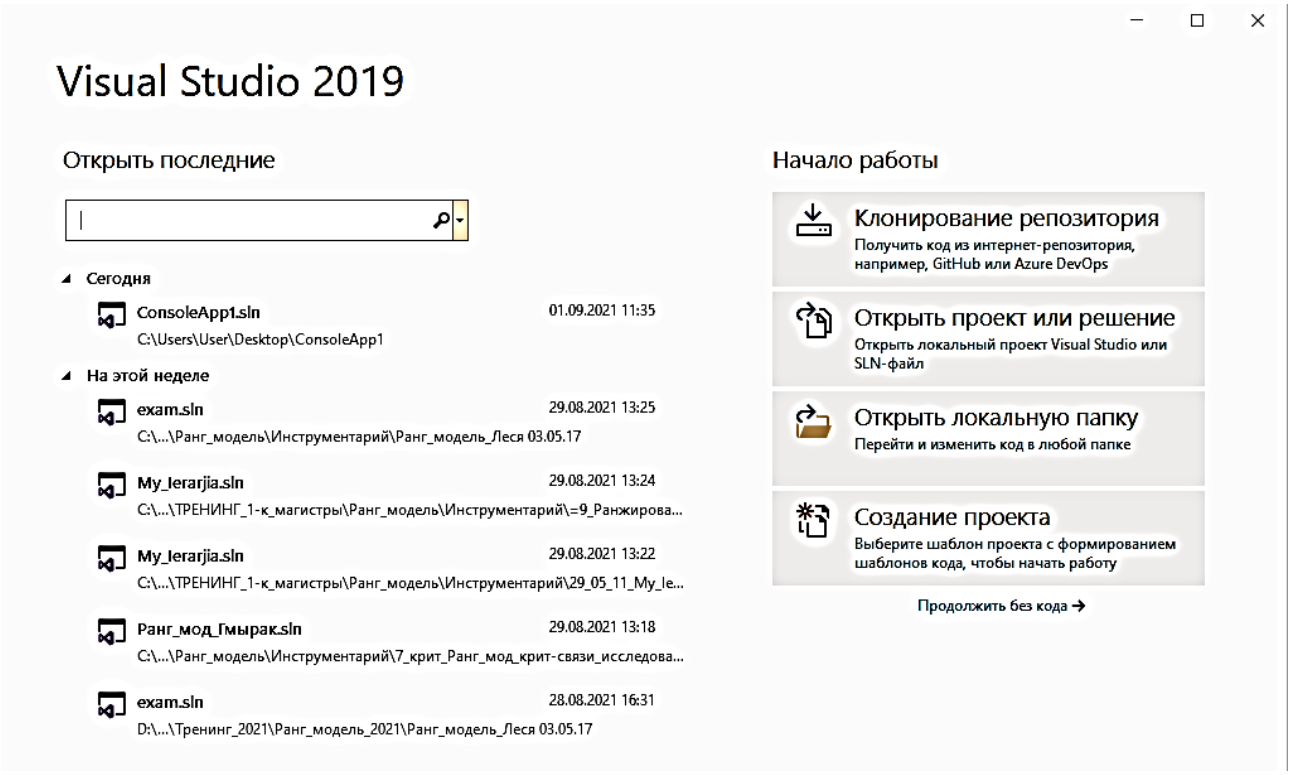


Рис. 4. **Основні частини візуального середовища розробки Visual Studio .NET (Ultimate 2012)**

Для версії Visual Studio .NET 2019 послідовність дій декілька інша. Нижче наведено перелік відповідних скріншотів, які пояснюють формування візуального середовища розробки.



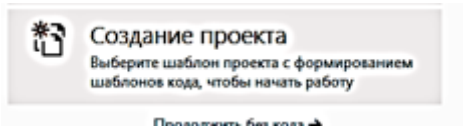
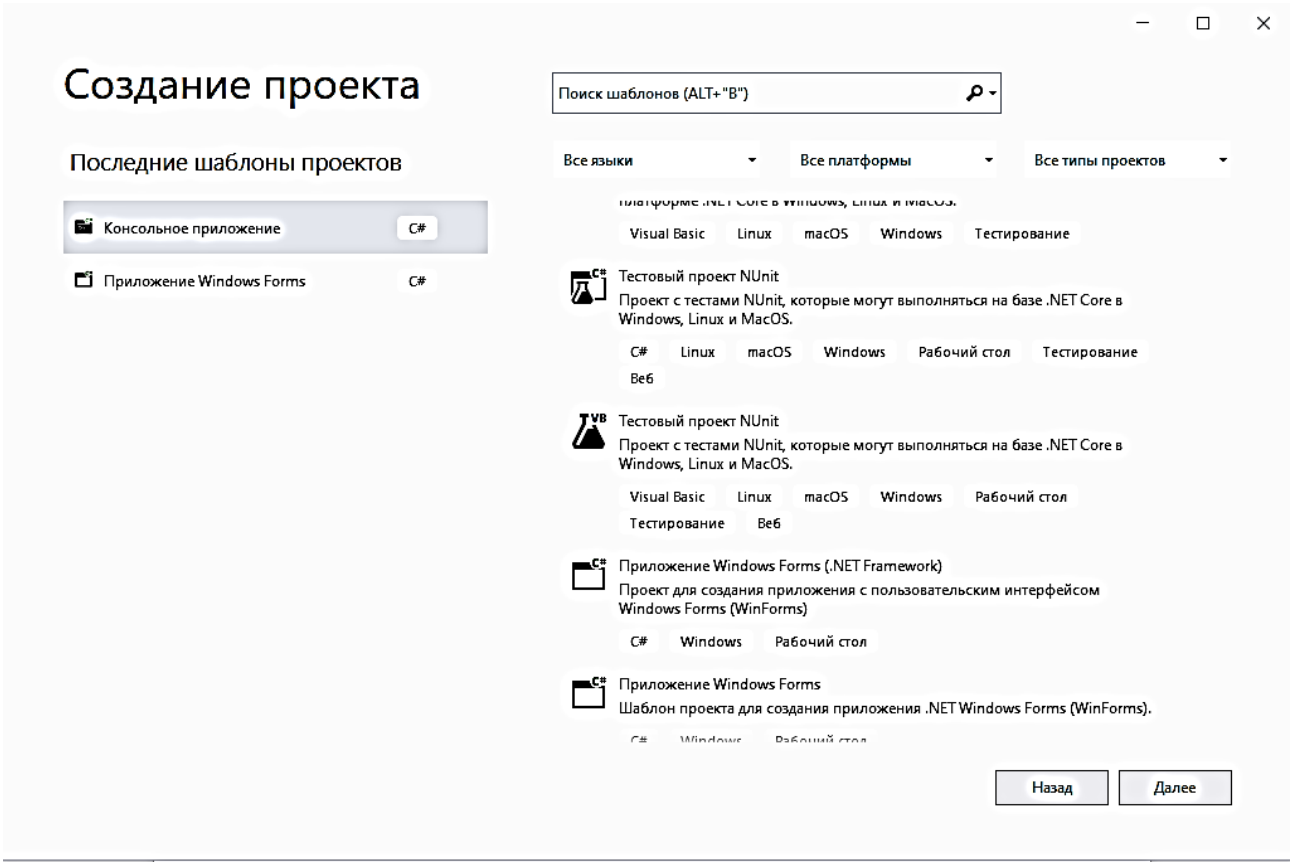


Рис. 4.1. **Стартова сторінка Visual Studio .NET 2019**



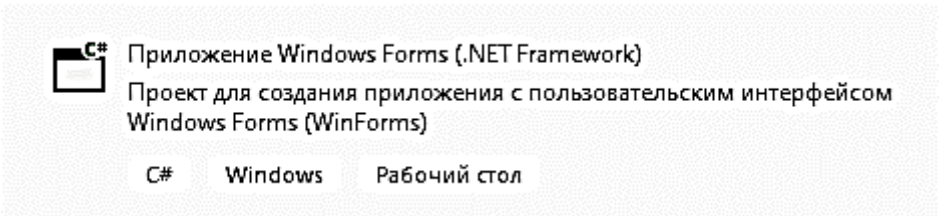


Рис. 4.2. **Вибір типу проекту (Visual Studio .NET 2019)**

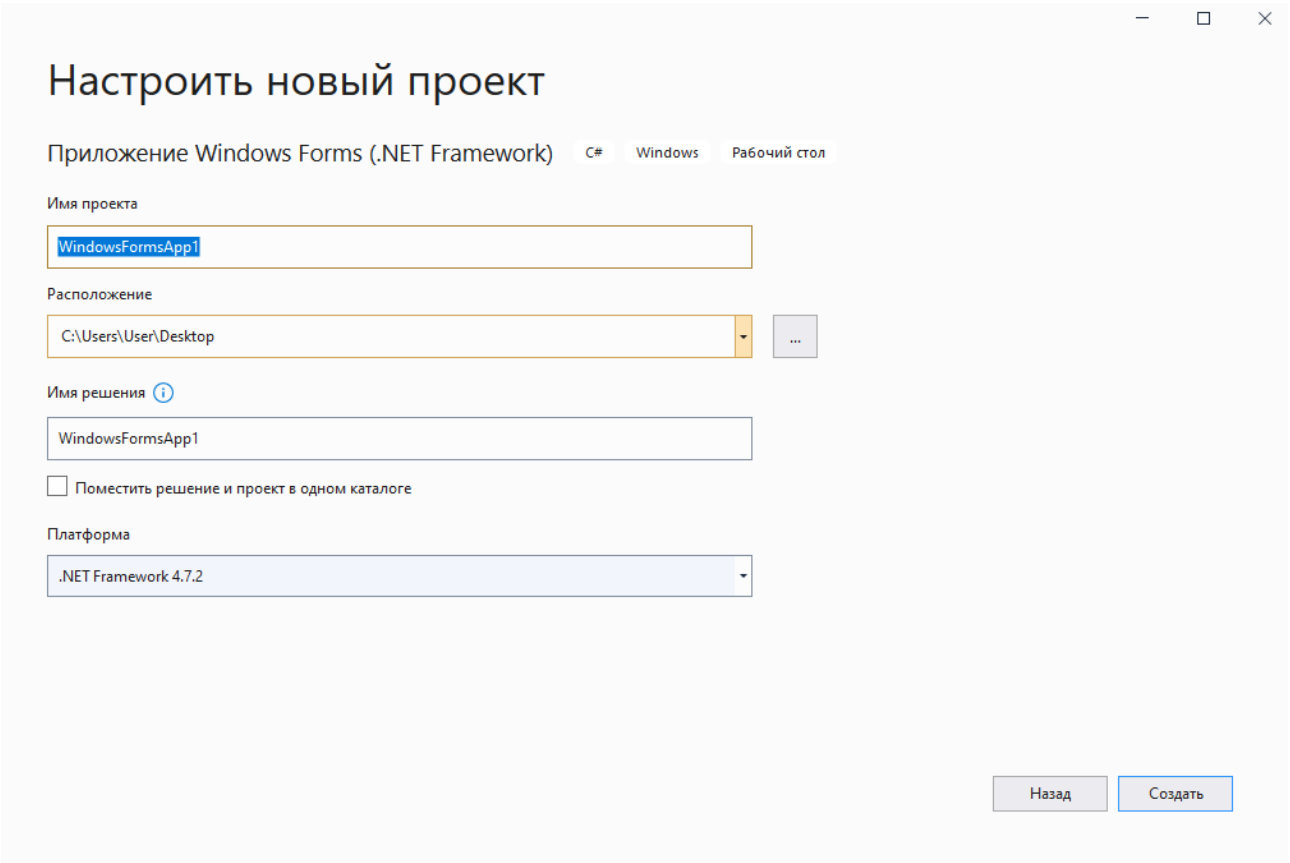


Рис. 4.3. **Вибір ім’я проекту та місця його розташування (Visual Studio .NET 2019)**

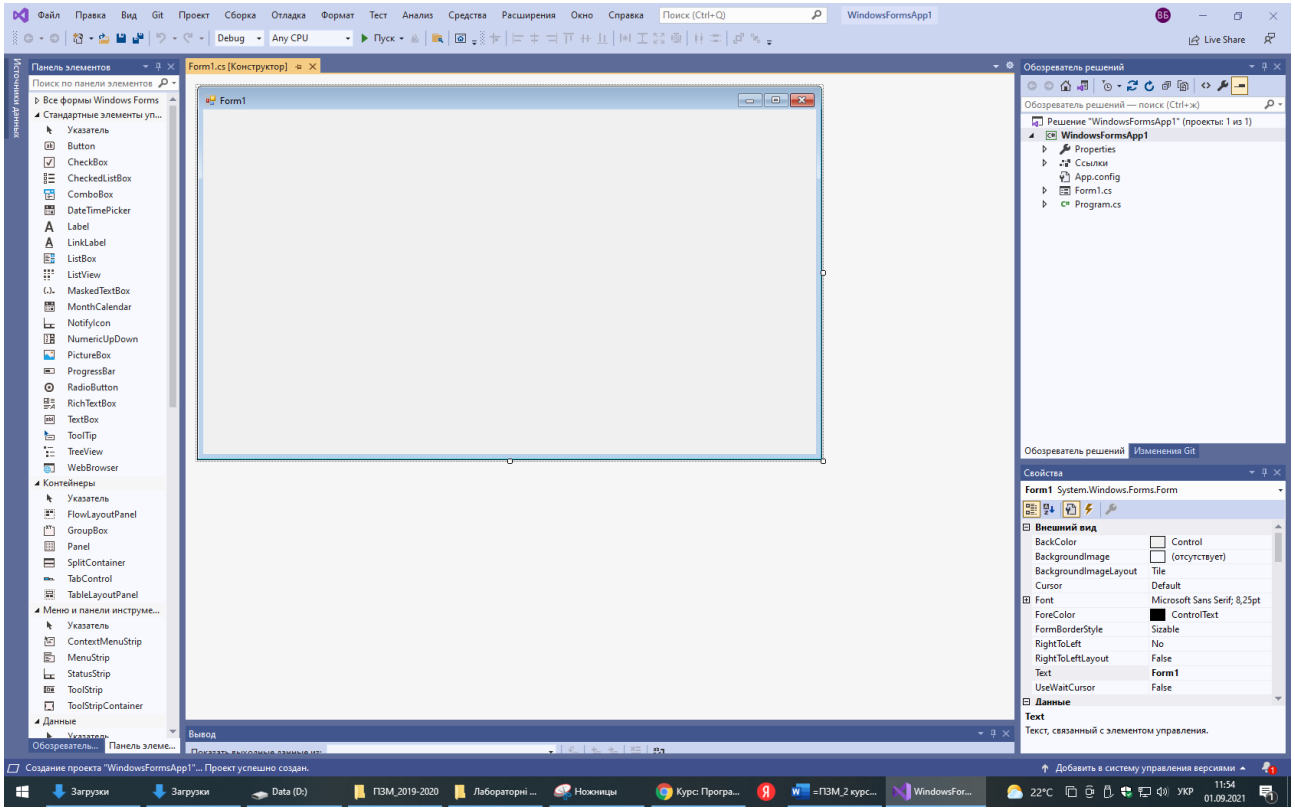


Рис. 4.4. **Основні частини візуального середовища розробки Visual Studio .NET 2019**

У центрі знаходиться головне вікно для створення візуальних форм і написання коду.

Праворуч розміщується вікно Solution Explorer для управління проектами та вікно властивостей Properties.

Solution Explorer дозволяє управляти компонентами, що входять до складу в проект.

Class View. Це вікно дозволяє переміщуватися по всіх елементах програмного проекту, включно окремими процедурами.

За допомогою Class View можна додавати нові методи, класи, дані.

Кожен елемент дерева проекту має контекстне меню. Якщо вікно Class View відсутнє на екрані, слід вибрати пункт меню View / Class View.

Properties Explorer. Це вікно дозволяє працювати з властивостями форм і їхніх компонентів. Воно містить список усіх властивостей обраного в поточний момент компонента. До того ж значення властивостей можуть бути подані в будь-якій формі, наприклад, як текстове поле, як допустимі значення, як вікно вибору кольору і т. д. Якщо змінити значення властивості за замовчуванням, то воно буде

виділено жирним кольором. У цьому випадку контроль за змінами, що вносяться в проект, стане більш наочним.

Другим важливим завданням, яке виконує Properties Explorer, є управління подіями. Для того щоб переключитися на закладку подій, слід натиснути кнопку з зображенням блискавки вгорі вікна.

Вікно подій дозволяє налаштовувати реакцію форми або компонента на різні дії з боку користувача або операційної системи, наприклад, створити обробник подій від миші або клавіатури. У лівій частині вікна міститься список всіх доступних подій, а в правій — імен методів, що обробляють події. За замовчуванням список методів порожній. Можна додати новий обробник, вписавши ім'я методу в відповідному полі, або створити обробник з ім'ям за замовчуванням, клацнувши два рази по комірці лівою кнопкою миші.

Toolbox. Це вікно містить Windows Forms компоненти, які можна розмістити на своїй формі. Якщо такого вікна у Visual Studio немає, виберіть у головному меню пункт View / Toolbox. Toolbox має кілька закладок. Наприклад, закладка All Windows Forms містить візуальні елементи управління такі, як кнопки, списки та дерева. Закладка Data присвячена базам даних. Закладка Components містить невізуальні компоненти, найбільш представницьким, серед яких є Timer.

Візуальні властивості допоміжних вікон. Усі візуальні вікна вони можуть “прилипати” до будь-якої сторони головного вікна Visual Studio .NET.

Візуальні вікна можуть ховатися під час втрати активності. Для того, щоб наділити цією властивістю, наприклад, Solution Explorer, слід вибрати у контекстному меню цього вікна пункт Auto Hide або натиснути відповідну кнопку поруч із кнопкою заголовка “Закрити“.

Щоб повернути вікно в первинний стан, варто просто клацнути лівою кнопкою миші по відповідній назві в панелі.

Меню і панель інструментів. Усі дії, які можна виконувати в середовищі Visual Studio .NET, розташовуються в головному меню. Головне меню має контекстну залежність від поточного стану середовища, тобто містить різні пункти залежно від того, чим зараз займається користувач і в якому вікні він знаходиться. Крім того, більшість пунктів меню продубльовано в панелі інструментів.

Visual Studio .NET має безліч панелей інструментів. Можна включити або виключити панель інструментів за допомогою меню View / Toolbars.

Ті панелі інструментів, які вже відкриті, позначені в меню “пташками“. Також можна створювати власні панелі інструментів, скориставшись пунктом цього ж меню Customize.

Головне меню Visual Studio .NET. Меню Visual Studio .NET знаходиться у верхній частині середовища. В меню є всі команди, призначені для виконання дій над елементами проектів. Пункти меню бувають командними та груповими (що містять інші пункти меню).

Назва кожного групового пункту меню відображає команди, що містяться в ньому. Наприклад, меню File містить команди, призначені для роботи з файлами проекту. Деякі пункти меню містять вкладені пункти з більш докладними командами. Наприклад, команда New з меню File показує меню вибору типів файлів. Найбільш часто вживані пункти меню мають «гарячі» клавіші. Так, для створення нового файла потрібно натиснути клавіші CTRL + N. Основні пункти головного меню Visual Studio .NET буде розглянуто далі.

Порядок виконання лабораторної роботи

Створення простого консольного додатку

Буде використано консольні додатки впродовж всього лабораторного практикуму першого семестру.

Процес створення консольного додатка складається з таких кроків.

1. Вибрати пункт меню File / New / Project.

2. У вікні New Project встановити відповідні (рис.5) налаштування: Visual C# / Windpws; тип проекту Console Application.

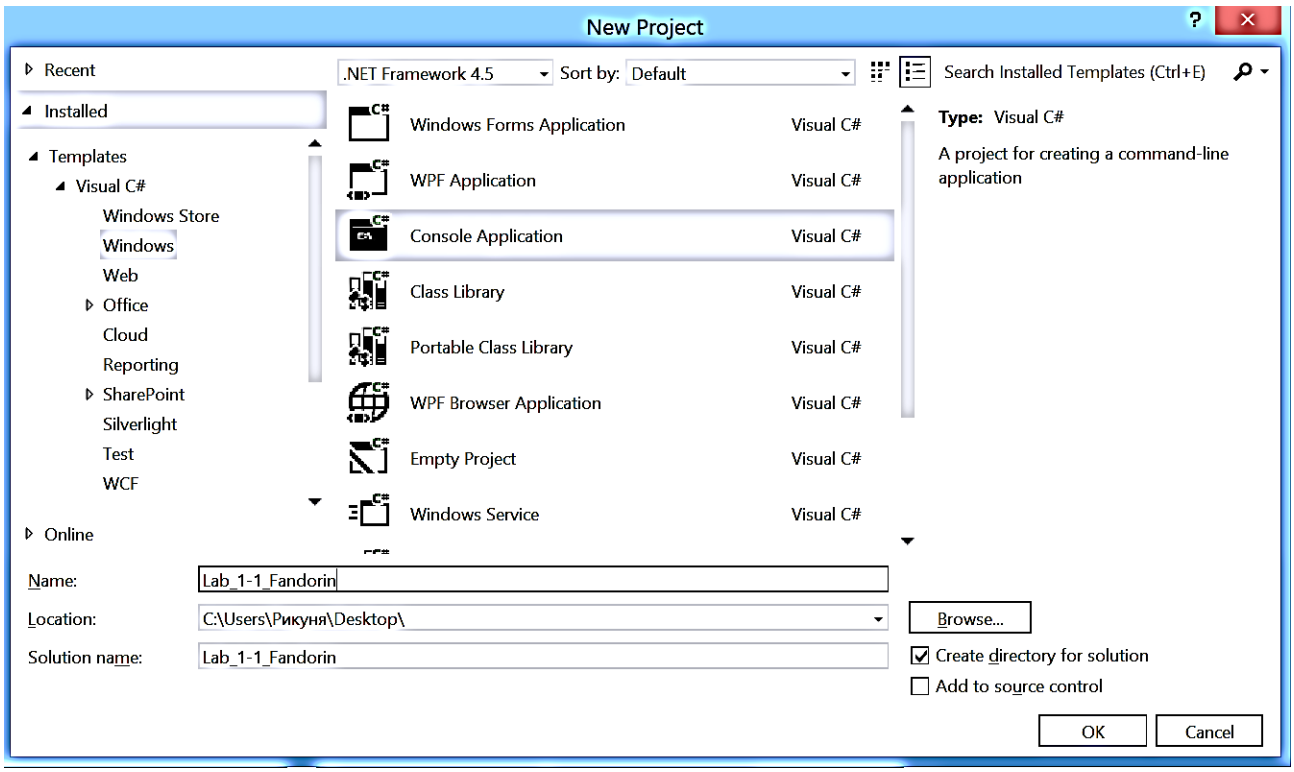


Рис. 5. **Вікно New Project для створення простого консольного додатка (середовище VS 2012)**

3. У полі Location вказати шлях до папки, в якій буде створено проект (якщо цієї папки не існує, вона буде створена автоматично).

4. У полі Name записати ім'я проекту (наприклад, Lab_1-1_Fandorin або залишити без змін, див. рис. 5).

4. Клацніть по кнопці ОК.

5. Після того, як шаблон проекту буде створено, додати в файл, виведений в основному вікні (рис. 6), рядок коду

Console.WriteLine("Моя перша програма!");

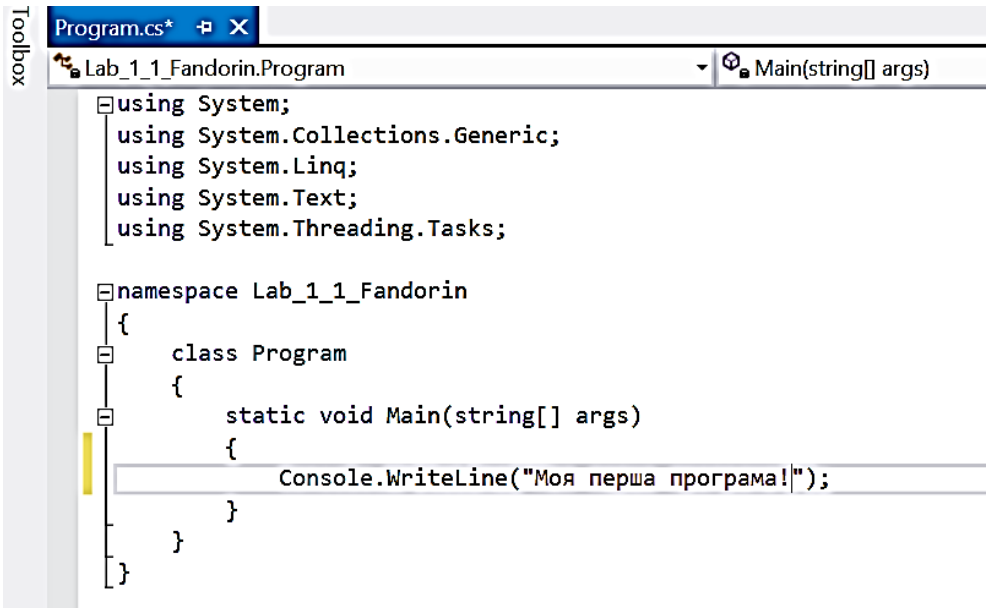


Рис. 6. **Перша консольна програма (середовище VS 2012)**

6. Вибрати пункт меню Debug / Start Without Debugging (або натиснути Ctrl + F5).

Через кілька секунд з'явиться наступне вікно (рис. 7).

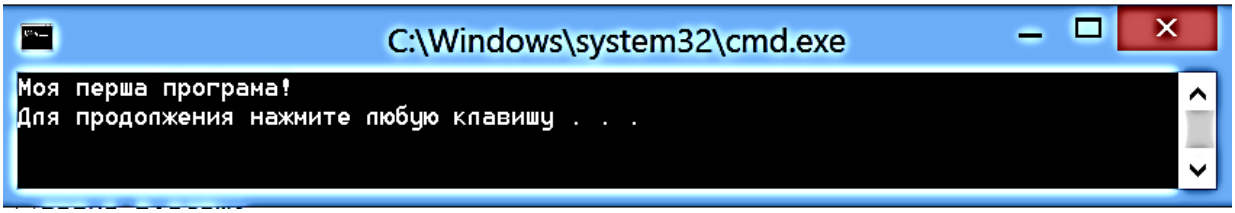
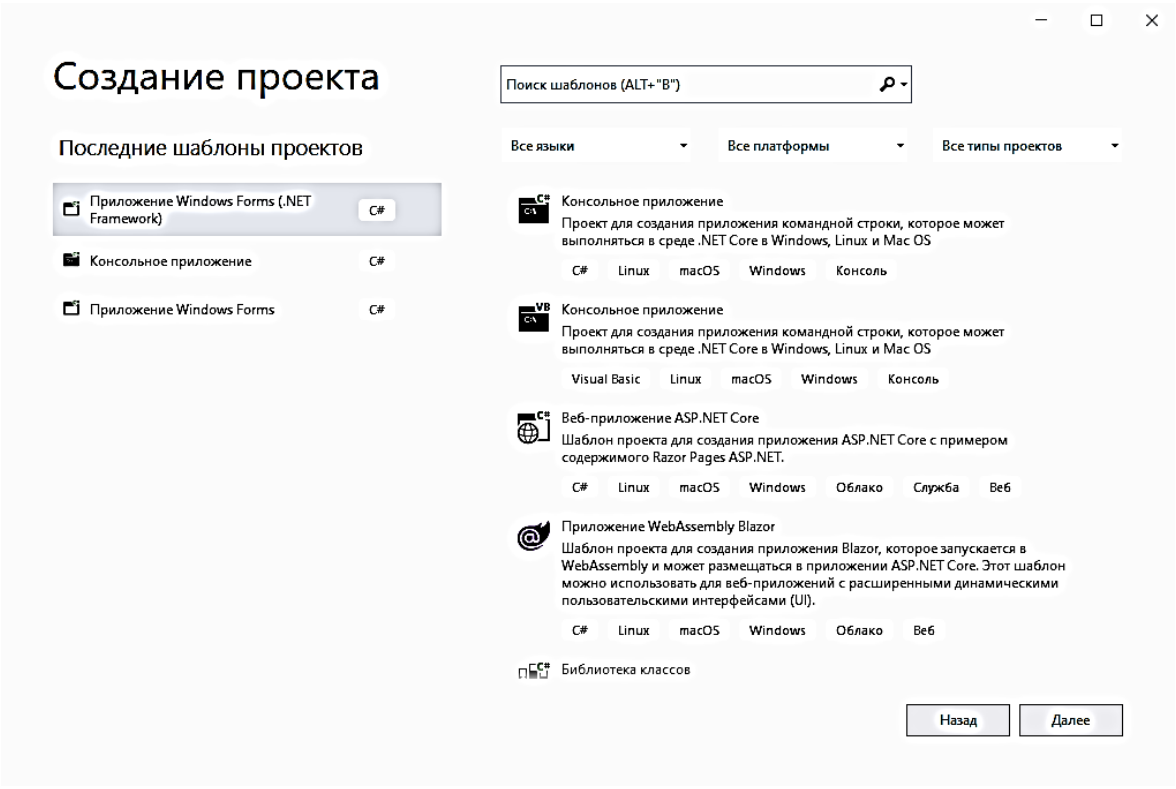


Рис. 7. **Результат виконання першої консольної програми (середовище VS 2012)**



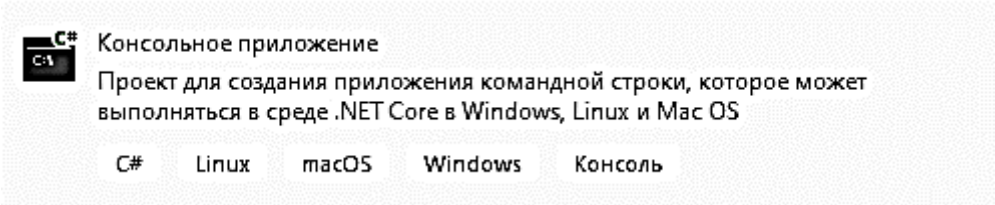
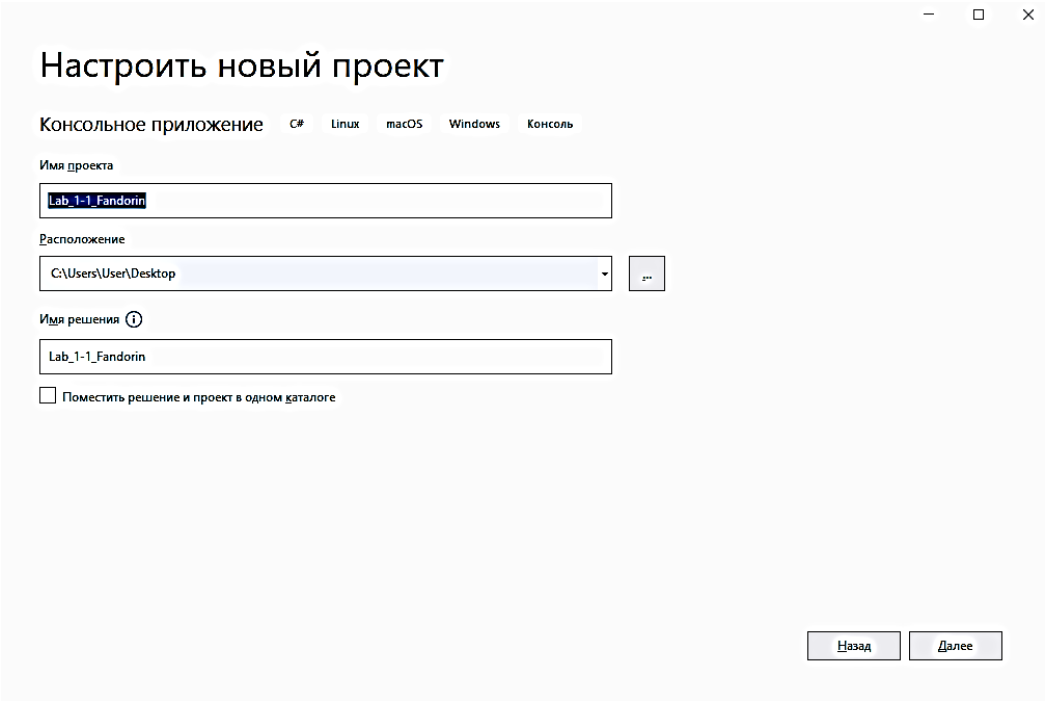
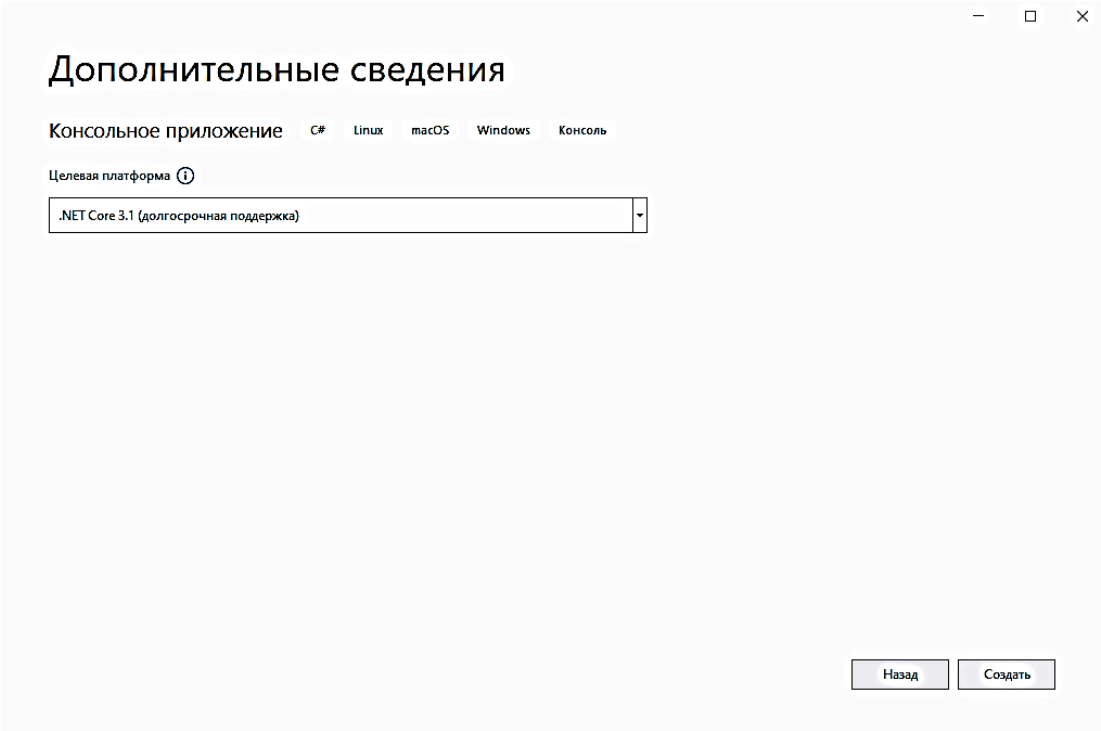


Рис. 7.1. **Вікно для створення простого консольного додатка (середовище VS 2019)**



Перше вікно



Друге вікно

Рис. 7.2. **Вікна для налаштування простого консольного додатка (середовище VS 2019)**

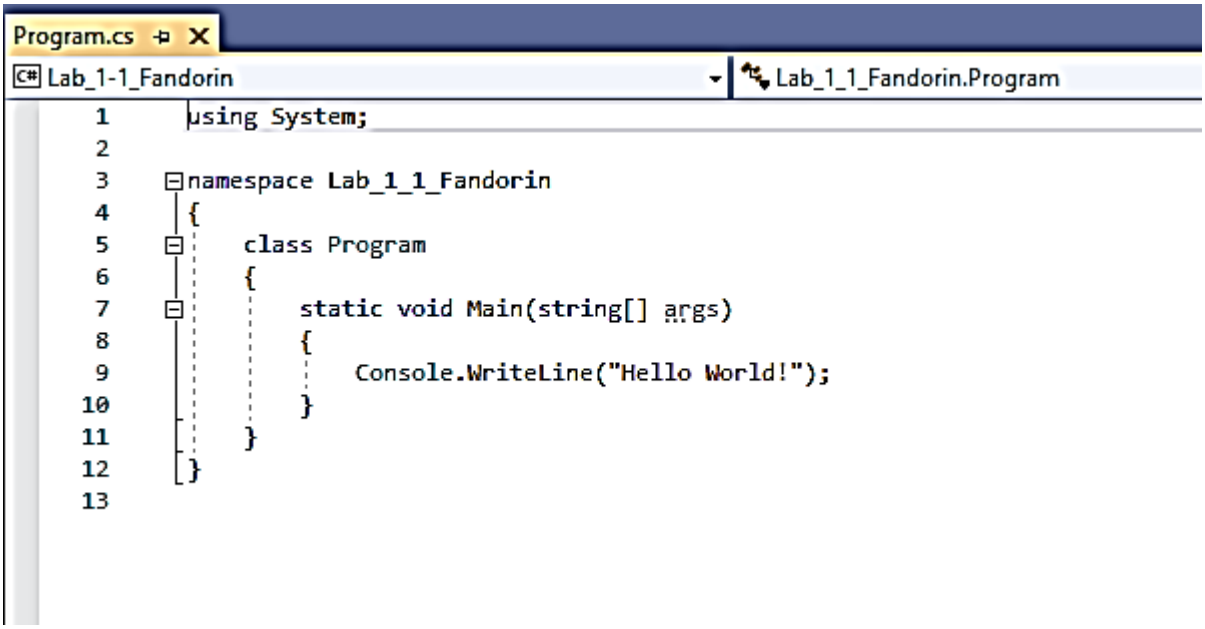


Рис. 7.3. **Перша консольна програма (середовище VS 2019)**

Рис. 7. 4. **Результат виконання першої консольної програми (середовище VS 2019)**

На цьому етапі не буде проаналізовано сам код, оскільки поки становить інтерес власне використання VS для введення та запуску програми.

Як можна помітити, VS робить величезний обсяг роботи, істотно спрощуючи процес компіляції та виконання коду. Однак навіть ці прості кроки можна виконувати різними способами.

Так, наприклад, створення нового проекту може бути здійснено за допомогою пункту меню File / New / Project (як було показано), за допомогою натиснення клавіш Ctrl + Shift + N або клацанням мишею по відповідному значку на панелі інструментів.

Ця програма також може бути відкомпільована та виконана кількома різними способами. Метод, який був використаний, вибравши пункт меню Debug / Start Without Debugging, має два спрощених способи виклику: з клавіатури (Ctrl + F5) і за допомогою іконки на панелі інструментів. Крім цього, існує можливість запускати код у режимі налагодження за допомогою пункту меню Start Debugging (той же результат можна отримати під час натискання клавіші F5). Після того, як процес компіляції закінчився, можна виконати згенерований файл із розширенням .exe, запустивши його з директорії, зазначеної (для розглянутого прикладу) на рис. 8.

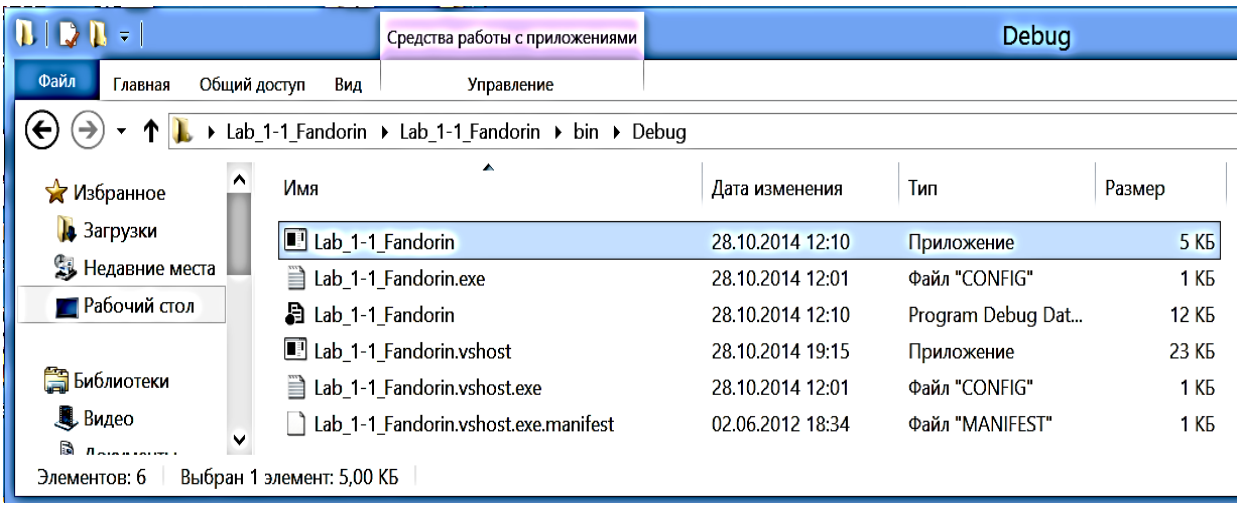
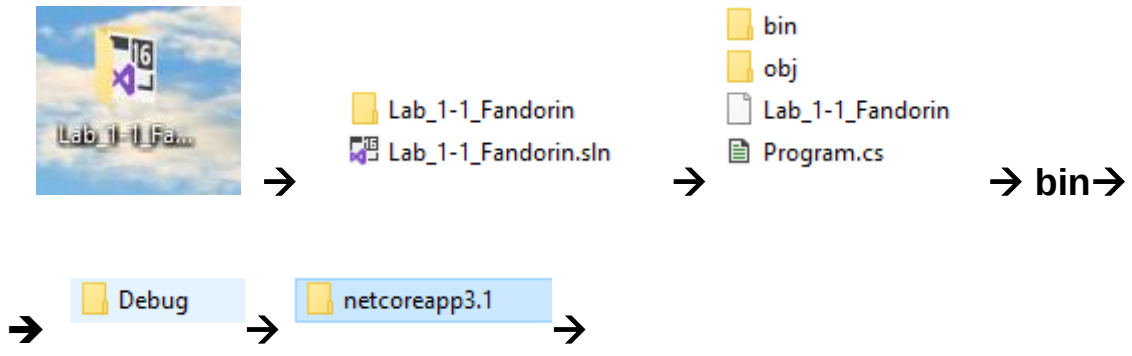


Рис. 8. **Директорія з файлами, які отримано в результаті компіляції проекту (середовище VS 2012)**



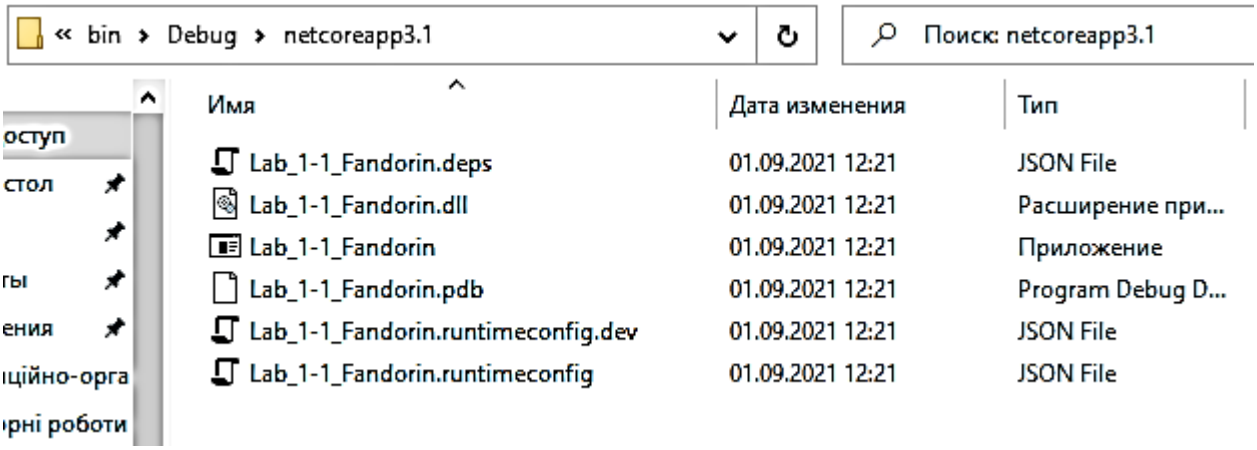


Рис. 8.1/ **Директорія з файлами, які отримано в результаті компіляції проекту (середовище VS 2019)**

Висновки

У загальних рисах розглянуто середовище розроблення Visual Studio .NET.

Показані приклади використання Visual Studio .NET для створення двох типів додатків. Більш простий з них — консольний додаток, його цілком достатньо для вирішення більшості завдань на початковому етапі вивчення мови C#, і він дозволяє зосередитися на основах програмування на C#.

Віконні додатки дещо складніші, проте візуально вони виявляються більш вражаючими та наочними для користувачів, знайомих з віконним середовищем.

Наступні лабораторні заняття присвячені основам синтаксису C# і структурі програм на C#. Цей матеріал необхідно освоїти, перш ніж переходити до складніших об'єктно-орієнтованих додатків.

Зміст звіту

1. Титульний лист.
2. Цілі лабораторного заняття та вказівка, які навички та вміння передбачається отримати в результаті його виконання.
3. Тексти налагоджених демонстраційних програм лабораторного заняття з необхідними коментарями та результатом виконання.
4. Висновки.

Контрольні запитання

1. Перерахуйте основні етапи розроблення програми, що виконується.

2. Навіщо необхідний етап компіляції?

3. У чому сутність процедурно-орієнтованого стилю програмування? Яка структура процедурно-орієнтованої програми?

4. Укажіть недоліки процедурного стилю програмування та шляхи їхнього подолання.

5. Дайте визначення об'єкта та наведіть приклад з описом його властивостей і поводження.

6. Навіщо необхідне повторне використання програмного забезпечення? Наведіть приклади повторного використання.

7. Призначення бібліотеки класів .NET Framework.

8. Опишіть можливі типи C#-додатків і сфери їхнього застосування.