**Лабораторна робота № 1**

Тема: Знайомство з візуальним середовищем програмування Microsoft Visua Studio і мовою програмування С#

Мета: Навчитись створювати проекти у середовищі Visual Studio.

**Теоретичні відомості**

**Платформа .NET Framework** - це кероване середовище, що надає різноманітні служби виконуваним в ній програмам. Воно складається з двох основних компонентів: Середовища CLR (ядро виконання, яке обробляє виконувані програми), і бібліотеки класів.NET Framework (яка надає бібліотеку перевіреного коду, який можна використовувати декілька разів і який можна викликати з програм). Служби, які платформа .NET Framework надає працюючим програмам, включають наступне:

* **Керування пам'яттю.** У багатьох мовах програмування програмісти самостійно призначають і виділяють ресурси пам'яті і вирішують питання, пов'язані з часом життя об'єктів. У програмах платформи .NET Framework середовище CLR надає ці служби від імені програми.
* **Система загального типу.** У традиційних мовах програмування, базові типи визначаються компілятором, що ускладнює взаємодію між мовами. В платформі .NET Framework базові типи визначаються системою типу .NET Framework. При цьому використовуються одні й ті ж базові типи для всіх мов .NET Framework.
* **Розширена бібліотека класів.** Замість того щоб писати багато коду для виконання стандартних низькорівневих операцій програмування, розробники можуть використовувати легкодоступну бібліотеку типів і члени з бібліотеки класів .NET Framework.
* **Платформи і технології розробки.** Платформа .NET Framework включає бібліотеки для конкретних областей розробки додатків, наприклад ASP.NET для веб-додатків, ADO.NET для доступу до даних і Windows Communication Foundation для програм, орієнтованих на служби.
* **Взаємодія мов.** Мовні компілятори, націлені на платформу .NET Framework, дають проміжний код, званий мовою CIL (Common Intermediate Language), який, у свою чергу, компілюється під час виконання середовищем CLR. За допомогою цієї функції, підпрограми, написані однією мовою, доступні в інших мовах, а розробники можуть зосередитися на створенні додатків на улюбленому мовою або мовами.
* **Сумісність версій.** За рідкісними винятками, програми, які розробляються за допомогою платформи .NET Framework певної версії, можуть виконуватися без змін на більш пізньої версії.
* **Паралельне виконання.** Платформа .NET Framework допомагає у вирішенні конфліктів версій, дозволяючи інсталяцію декількох версій середовища CLR на одному комп'ютері. Це означає, що кілька версій програм також можуть співіснувати, і що програма може виконуватися на версії платформи .NET Framework, для якої вона була створена.
* **Налаштування для різних версій.** Орієнтуючись на переносиму бібліотеку класів платформи .NET Framework, розробники можуть створювати код, який працює на декількох платформах .NET Framework, наприклад .NET Framework, Silverlight, Windows Phone 7 або Xbox 360.

**Створення програмного забезпечення у середовищі Microsoft Visual Studio (на прикладі Microsoft Visual Studio 2010)**

Розглянемо для початку можливості даного середовища на прикладі створення консольного додатку.

Оскільки консольні програми виконують введення і виведення даних через командний рядок, вони ідеально підходять для швидкого ознайомлення з можливостями мови та написання службових програм командного рядка.

**Приклад 1.1. Консольний додаток Hello World!**

1. Запустіть середовище Microsoft Visual Studio. У меню Файл виберіть команду Створити проект (рис.1.1).

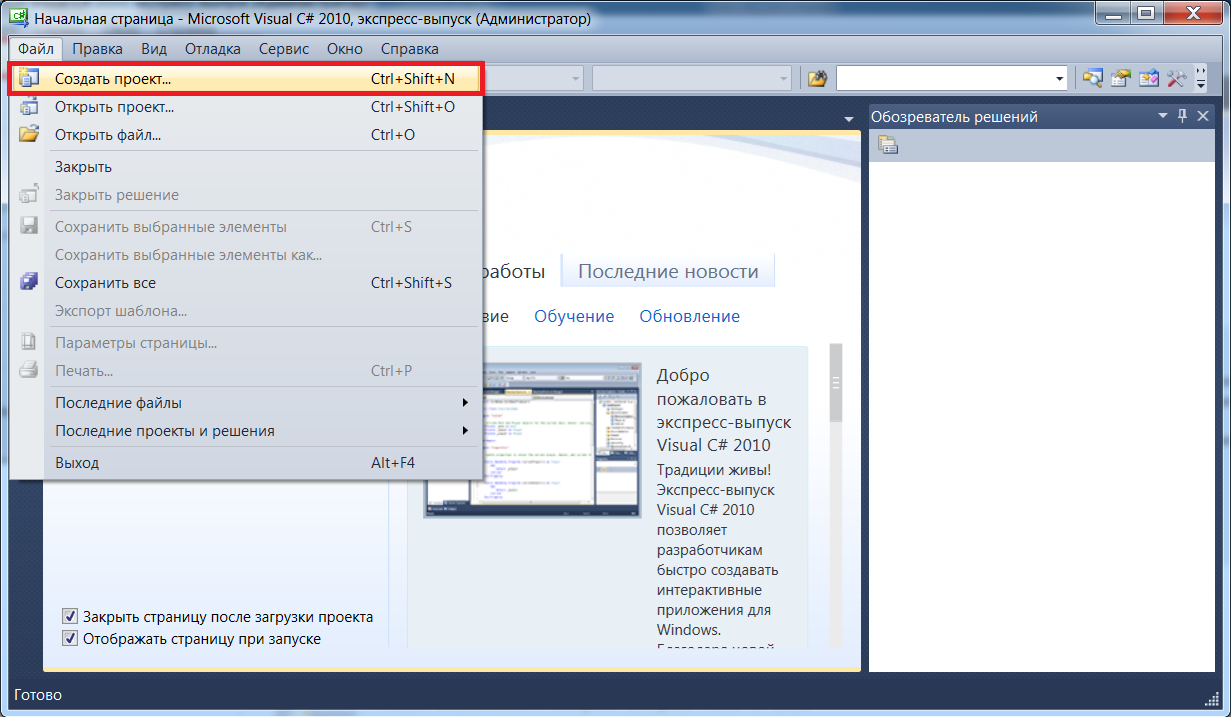


Рисунок 1.1 – Створення нового проекту у середовищі Visual Studio

1. Відкриється діалогове вікно Створення проекту. У цьому діалоговому вікні виводиться список різних типів додатків за замовчуванням, які можна створювати за допомогою Visual C#, експрес-випуск.

В якості типу проекту виберіть Консольну програму та відредагуйте ім'я додатка (рис. 1.2). Можна залишити розташування проекту за замовчуванням або вказати новий шлях на свій розсуд.

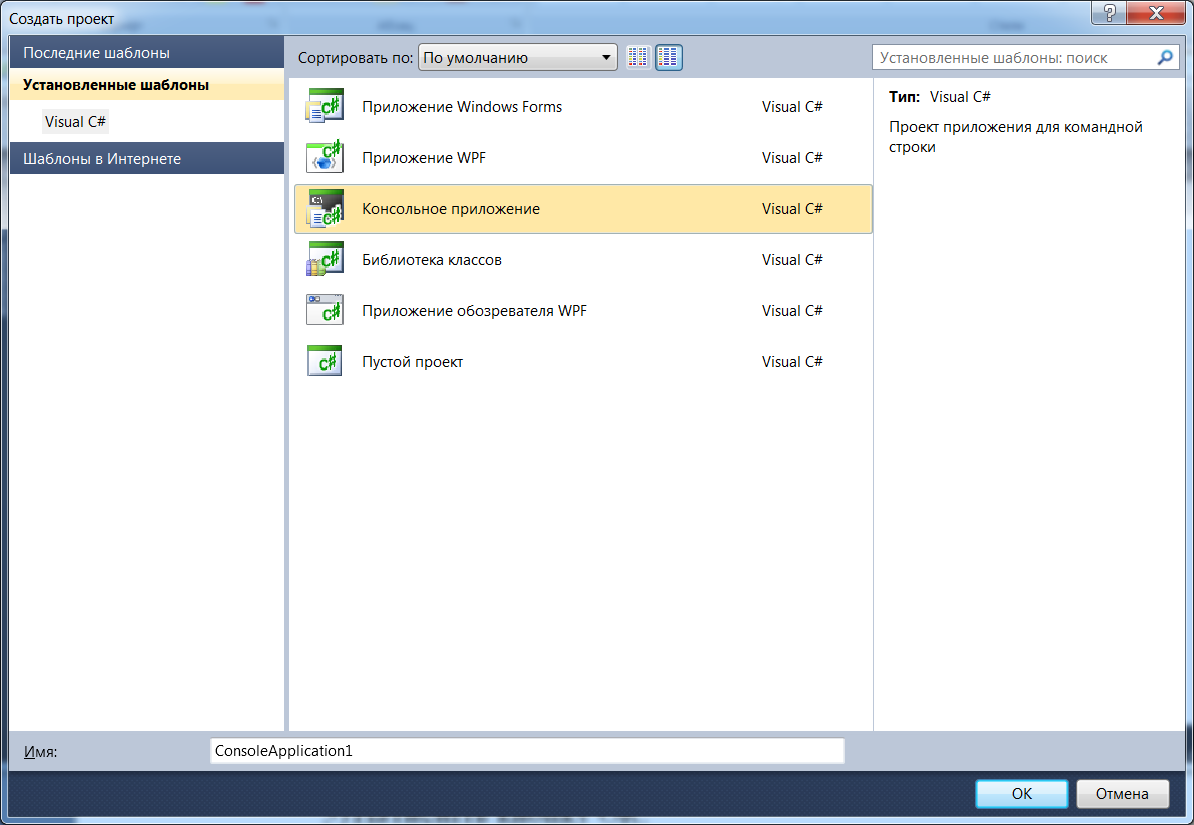


Рисунок 1.2 – Вибір типу програми

1. Натисніть кнопку ОК.

Visual C#, експрес-випуск створить нову папку для проекту з таким же ім'ям, як у проекту. Відкриється головне вікно Visual C#, експрес-випуск і панель коду для введення і зміни вихідного коду програми C#. (рис. 1.3)

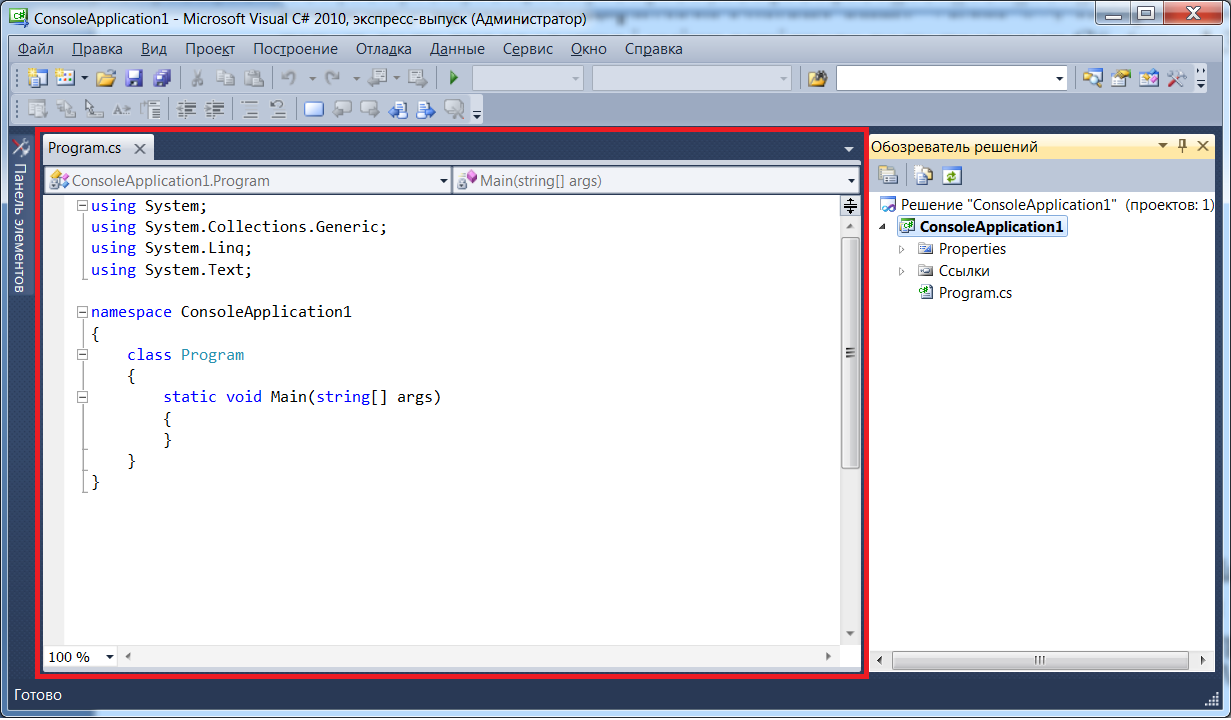


Рисунок 1.3 – Вікно редактора коду (позначене на рисунку червоним прямокутником)

Зверніть увагу на панель інструментів у верхній частині вікна. Панель інструментів містить значки для створення, завантаження та збереження проектів, редагування вихідного коду, побудови програми, приховування та відображення інших вікон, що є частиною середовища Visual C#, експрес-випуск. На правому краю панелі інструментів знаходяться п'ять значків для відкриття важливих вікон, таких як Оглядач рішень і Панель елементів. При наведенні покажчика миші на будь-який з цих значків з'являється вспливаюча підказка.

1. Переконайтеся, що відображається Оглядач рішень, клацнувши вкладку Оглядач рішень в правій частині екрана або значок Оглядач рішень в панелі інструментів (рис. 1.4).

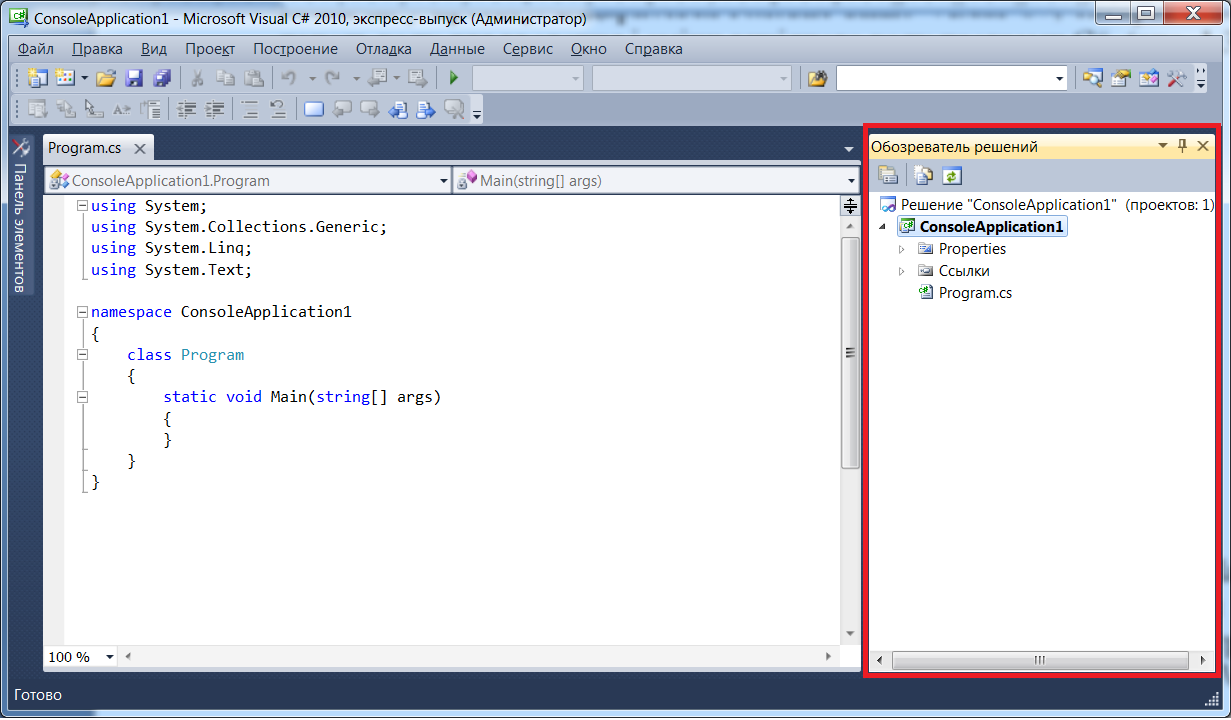


Рисунок 1.4 – Вікно оглядача рішень (позначене на рисунку червоним прямокутником)

Панель Оглядач рішень є дуже корисною, на ній відображаються різні файли, які є частиною проекту. Найважливішим файлом проекту є "Program.cs", який містить вихідний код програми.

Важливо знати, як відкривати і приховувати вікна, подібні Оглядаче вірішень, це дозволить зберігати Visual C# в належному вигляді. Оглядач рішень за замовчуванням є видимим. Щоб приховати Оглядача рішень клацніть значок Автоприховування (значок канцелярської кнопки в рядку заголовка) або відкрийте меню Параметри в рядку заголовка Оглядача рішень і виберіть Автоприховування. Інші вікна, такі як Представлення класів і Властивості, також мають ці значки.

1. Введіть ім'я класу **Console** в редакторі коду.

Якщо оглядач рішень, як і раніше приховує панель Код, клацніть панель Код для його приховування. Тепер клацніть праву відкриту фігурну дужку ({) всередині методу Main і натисніть ВВЕДЕННЯ для переходу на новий рядок. Зверніть увагу, що редактор робить відступи автоматично.

Редактор коду завжди намагається форматувати введений код в стандартній, зручній для сприйняття структурі. Якщо код виглядає непривабливо, можна переформатувати весь документ, вибравши пункти Додатково та Форматувати документ в меню Правка або натиснувши клавіші CTRL + E, D.

При введенні імені класу або ключового слова C# є вибір: або закінчити введення слова самостійно, або дозволити зробити це вбудованому в панель Код інструменту IntelliSense. Наприклад, при введенні символу "c" з'явиться вспливаючий список слів, пропонованих IntelliSense, з якого можна вибрати слово, що набирається. У цьому випадку, слово "Console" поки ще не видно. Або прокрутіть вниз список, або продовжуйте вводити слово "console". Коли слово "console" буде виділено в списку, натисніть клавішу ENTER, або TAB, або двічі клацніть його мишею, Console буде додано в код (рис. 1.5).

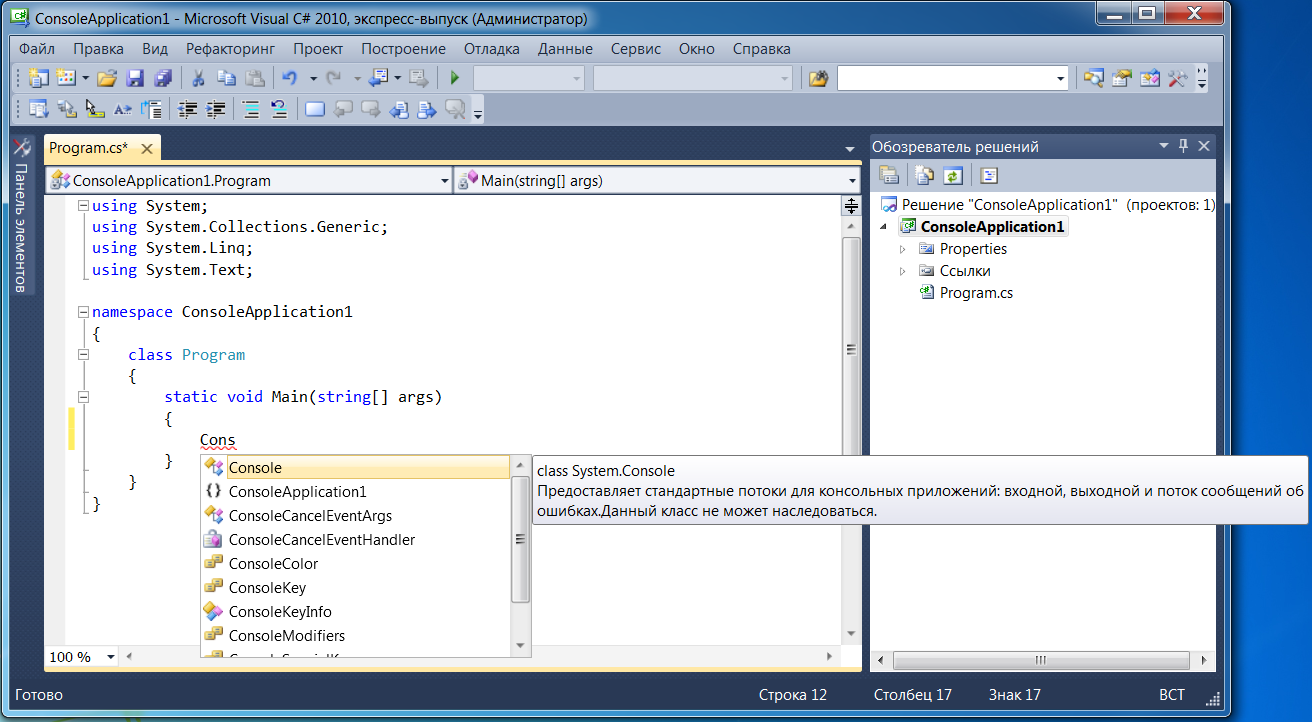


Рисунок 1.5 – Використання Intellisence

Перевагою використання IntelliSense є впевненість у правильності вибору регістра і написання слова. Користувач може або сам вводити код, або використовувати технологію IntelliSense.

1. Введіть точку і ім'я методу **WriteLine**.

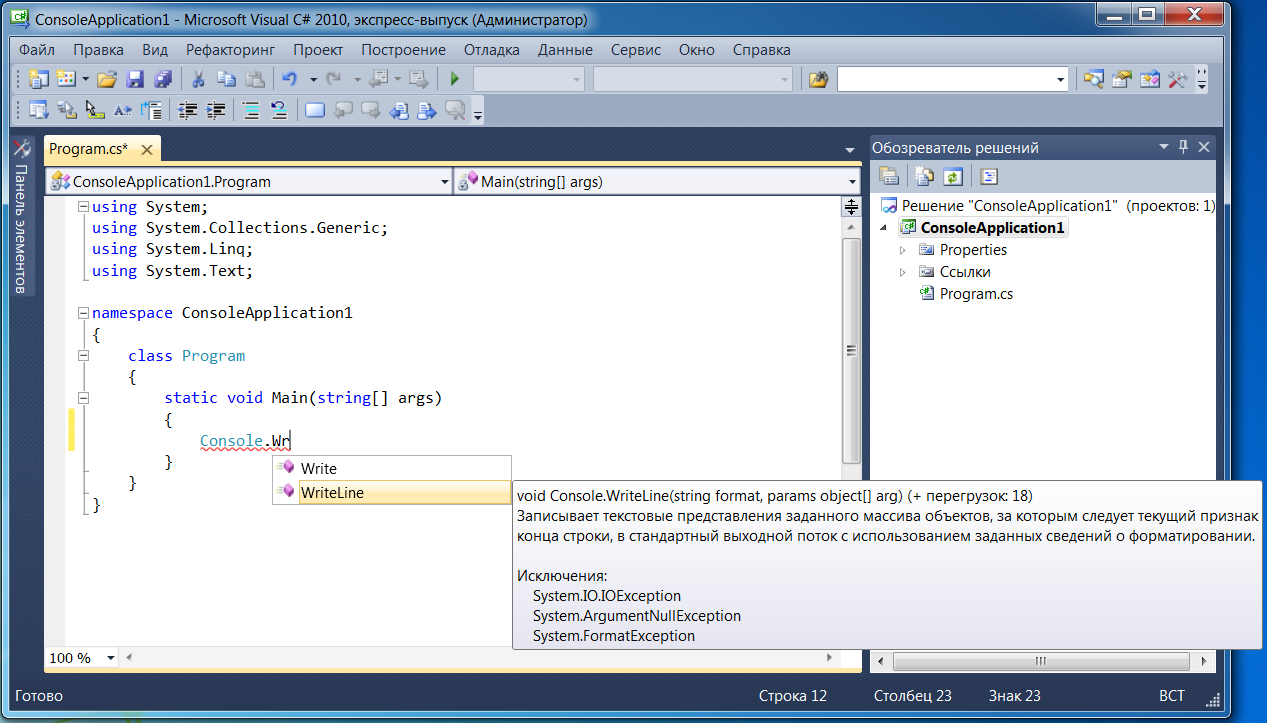


Рисунок 1.6 – Використання Intellisenceдля вибору методу класу

Як тільки користувач введе точку після Console, IntelliSense відобразить інший список. Список містить імена всіх можливих методів і властивостей, що є частиною класу Console. Необхідний метод WriteLine можна побачити вкінці списку. Або закінчіть введення слова WriteLine, або натисніть клавішу СТРІЛКА ВНИЗ для вибору, потім ENTER, або TAB, або двічі клацніть його мишею. Метод WriteLine буде додано до коду (рис. 1.6).

Введіть відкриваючу дужку. З'явиться повідомлення у вигляді спливаючої підказки з надписами методів, що є ще однією функцією IntelliSense. У цьому випадку буде відображено 19 різних надписів, які можна переглянути за допомогою клавіш СТРІЛКА ВГОРУ та СТРІЛКА ВНИЗ.

1. Введіть або вставте наступний код:

static void Main(string[] args)

{

Console.WriteLine("Hello World!");

// Зберегти вікно консолі відкритим до натиснення будь-якої клавіші

Console.WriteLine("Press any key to exit.");

Console.ReadKey();

}

Останній рядок у програмі Console.ReadKey() призначений для призупинення виконання програми до натискання будь-якої клавіші. Якщо не додати цей рядок, вікно з командним рядком відразу ж никне і користувач не зможе побачити виведення результату виконання програми. Якщо створювана службова програма завжди буде використовуватися в консолі командного рядка, виклик методу Console.ReadKey() можна опустити.

1. Виконайте програму.

Тепер програма завершена, готова для компіляції і виконання. Для цього або натисніть клавішу F5, або скориставшись кнопкою Пуск на панелі інструментів (рис. 1.7).

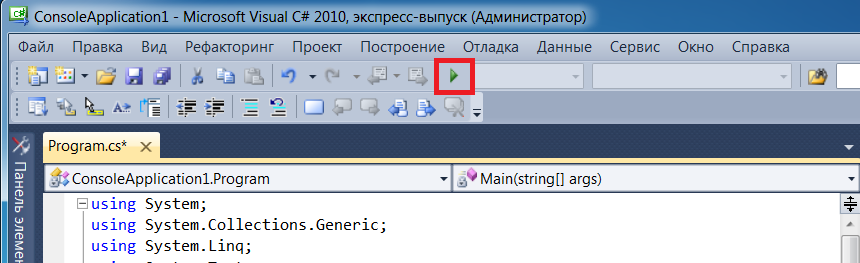


Рисунок 1.7 – Кнопка «Запуск» (виділена червоним прямокутником)

1. Після компіляції і виконання відкриється вікно консолі (рис. 1.8).

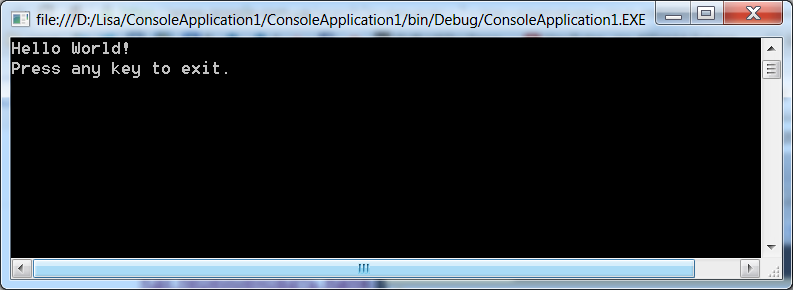


Рисунок 1.8 – Результат компіляції та виконання програми

**Приклад 1.2. Windows-додаток простий калькулятор**

Розглянемо приклад створення простого калькулятора (з графічним інтерфейсом користувача) в середовищі VisualStudio.

Послідовність дій для створення калькулятора:

1) Натисніть Файл → Створити проект... Відкриється вікно, а якому треба вибрати WindowsFormsApplication

2) На формі, що відкриється після попередньої дії, слід розмістити 3 текстових поля (об’єкт TextBox), та 5 кнопок (Button). Об’єкти, які можна розмістити на формі знаходяться в панелі ToolBox (рис. 1.9).

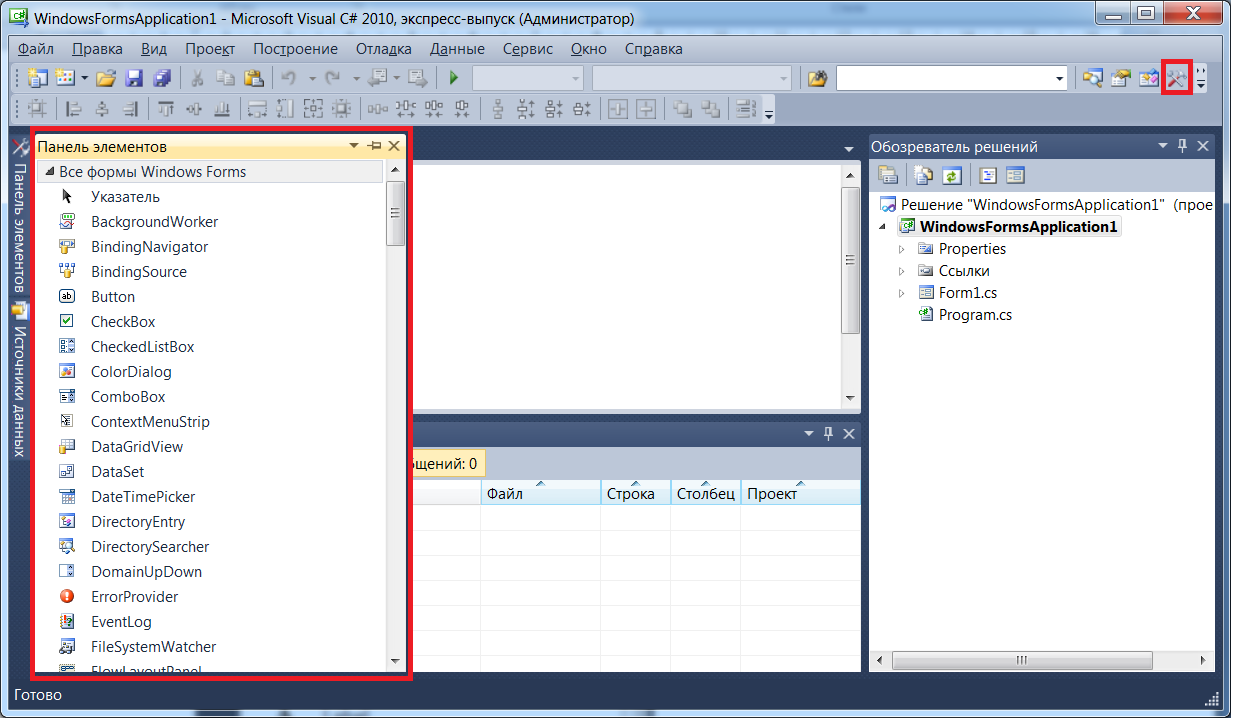


 Рисунок 1.9 – Панель ToolBox *(виділена червоним прямокутником)*

3) У кнопок треба змінити властивість Text на наступні значення:

* У button1 властивість Text –"CE";
* У button2 властивість Text –"\*";
* У button3 властивість Text –"/";
* У button4 властивість Text –"–";
* У button5 властивість Text –"+".

Властивості змінюються на панелі властивостей, яка виглядає наступним чином:

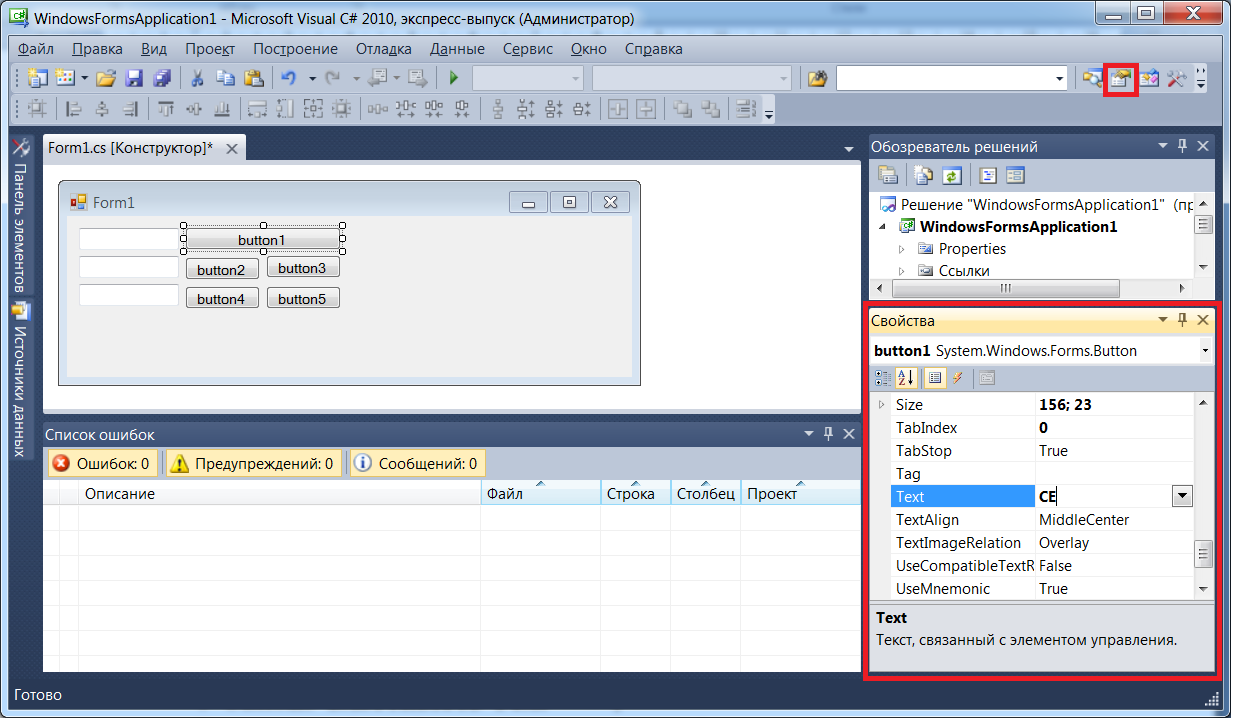


Рисунок 1.9 – Панель властивостей (виділена червоним прямокутником)

Після зазначених дій повинне вийти наступне зображення:

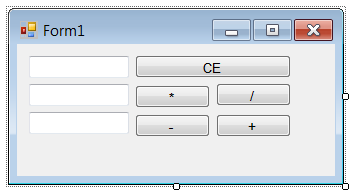


Рисунок 1.9 – Зовнішній вигляд програми Калькулятор

4) Клацніть два рази на створеній кнопці "\*" (множення). Відкриється вікно Редактора коду, в якому курсор миші буде розташований всередині функції, що обробляє натиснення на кнопку "\*". Додайте туди наступний код:

double result = Convert.ToDouble(textBox1.Text) \* Convert.ToDouble(textBox2.Text);

В цьому коді до змінної resultз типом doubleпривласнюється значення текстових полів, які треба помножити між собою.

Щоб одержати значення текстового поля пишемо ім'я текстового поля (textBox1), і властивість Text(textBox1.Text).

Але чому ми не написали так: textBox1.Text \* textBox1.Text ?

А тому що в текстовому полі зберігатися текст, а при множенні потрібно використовувати числа.

Тому за допомогою функції Convert.ToDouble ми міняємо тип змінної зі строкової в числову (із плаваючою точкою, тому що можливо доведеться множити дроби).

Далі результат, що зберігатися в змінній resultзаписуємо в текстове поле 3, при цьому міняємо тип змінної назад з числового в строковий:

textBox3.Text = result.ToString();

5) Приклад обробки помилки введення:

private void button2\_Click(objectsender, EventArgs e)

{

    // Якщо нічого не введено

    if (textBox1.Text == "")

       MessageBox.Show("Данні не введені");

    else

    {

       ...//здійснюємо множення

}

}

6) Для очистки текстових полів калькулятора застосуємо наступний код:

//очищаємо текстові поля (кнопка "СЕ")

private void button1\_Click(object sender, EventArgs e)

   {

   textBox1.Text = "";

   textBox2.Text = "";

   textBox3.Text = "";

   textBox4.Text = "";

   }

**Завдання до лабораторноїроботи:**

Створити програму "Калькулятор" на мові C# з графічним інтерфейсом, яка повинна виконувати чотири базові арифметичні дії (+, -, \*, /), а також операції х2, х! та обчислення остачі від ділення (%).

***Контрольнізапитання:***

1. Опишіть особливості платформи .NetFramework 4.5. Її переваги та недоліки в порівнянні з іншими платформами.
2. Які інші платформи для роботи з візуальними мовами програмування ви знаєте?
3. Як створити програму у середовищі Visual Studio?