

Лабораторна робота №1

Базовий синтаксис мови C#. Створення простих консольних застосунків у Visual Studio

Мета

Ознайомитися з базовими можливостями мови програмування C#, оволодіти використанням середовища Visual Studio для створення консольних додатків мовою C#. Вивчити принципи об'єднання кількох проєктів у рішення, ознайомитися з типами проєктів «консольний додаток» та «бібліотека класів». Отримати базову інформацію про класи та екземпляри класів.

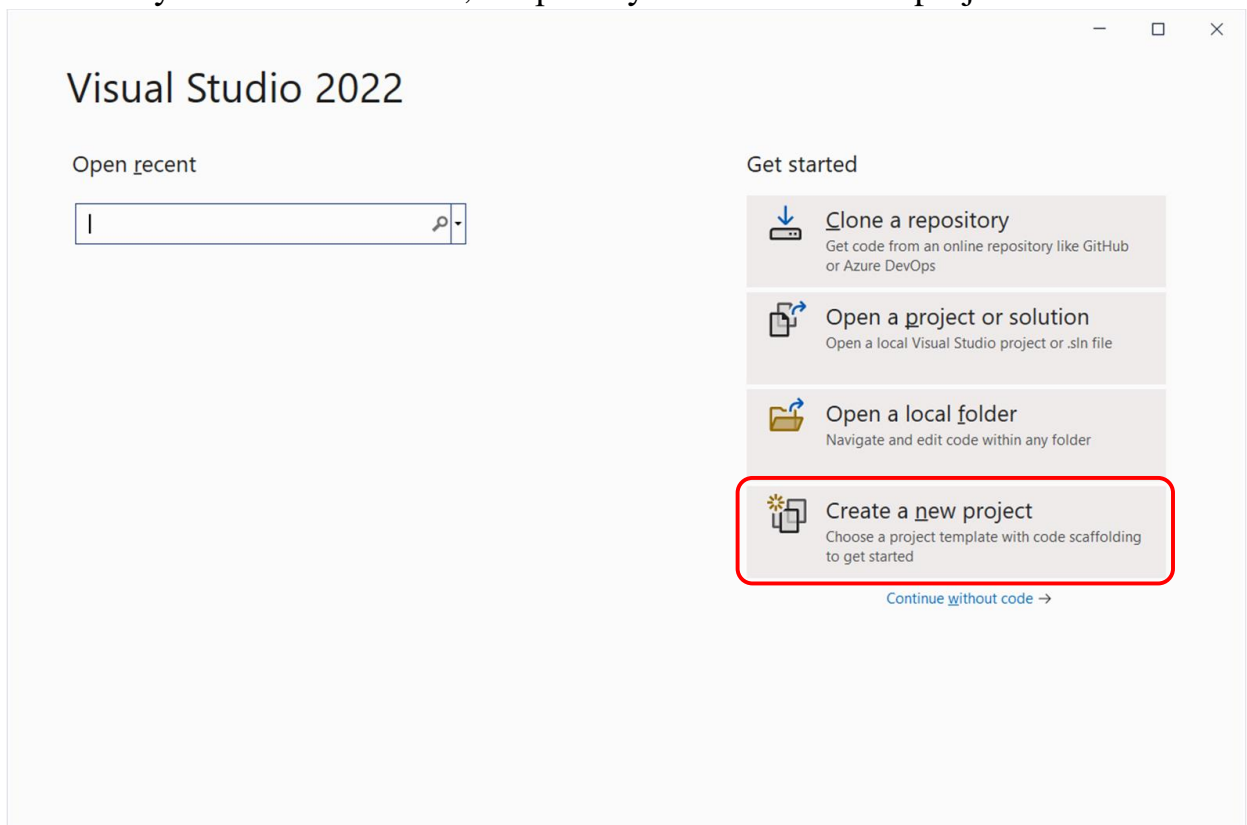
Завдання 1. Створення рішення та простого проєкту у Visual Studio

Теоретичні відомості

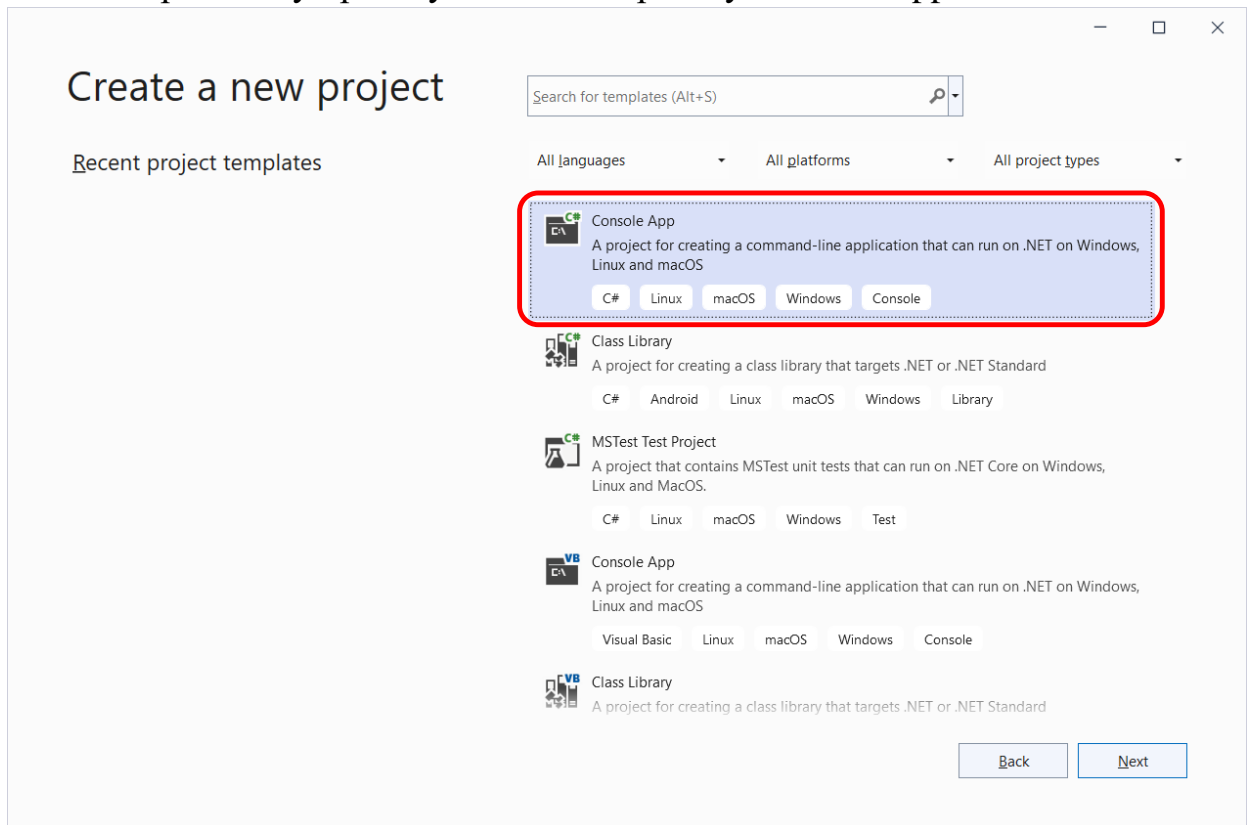
Рішення у Visual Studio – це контейнер високого рівня для інших елементів, таких як проєкти, файли readme, діаграми і т.д.

Послідовність виконання завдання

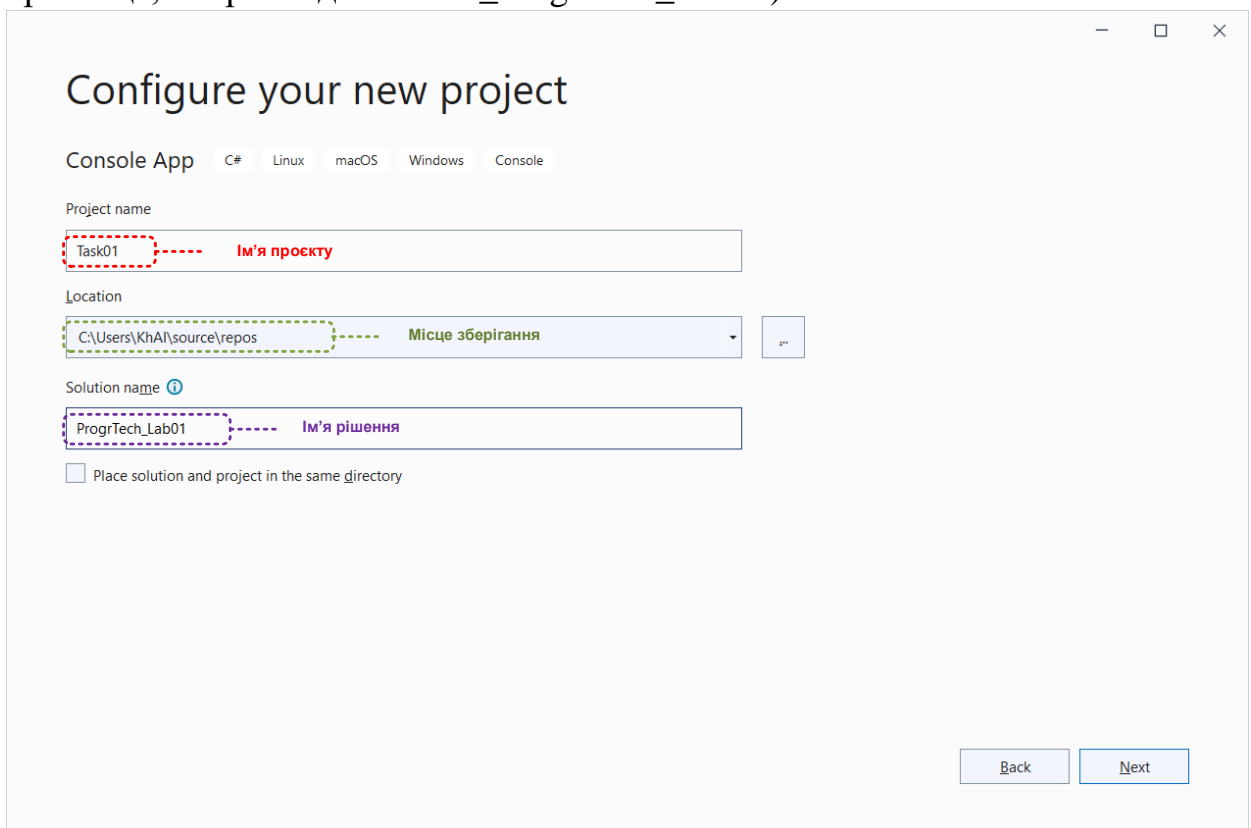
Запустіть Visual Studio, оберіть пункт Create a new project:



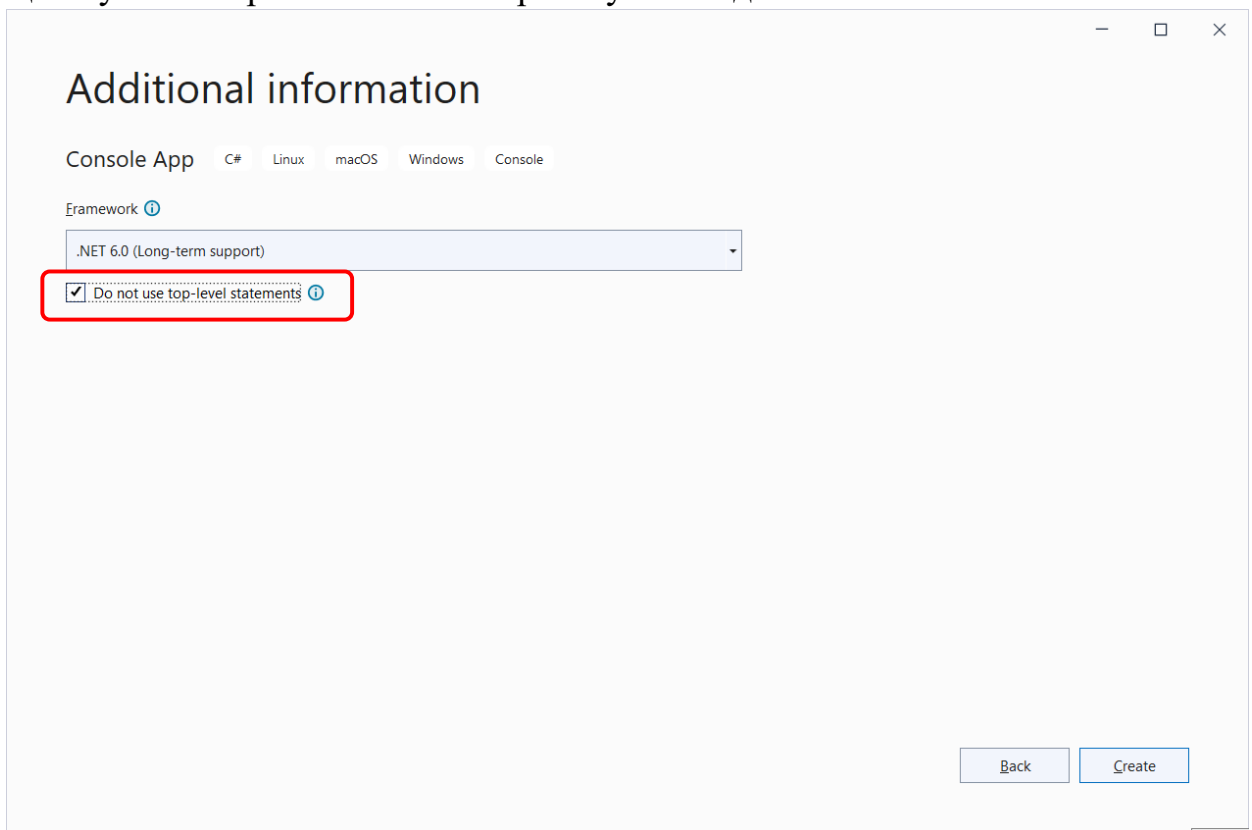
Оберіть мову проєкту C# та тип проєкту Console App:



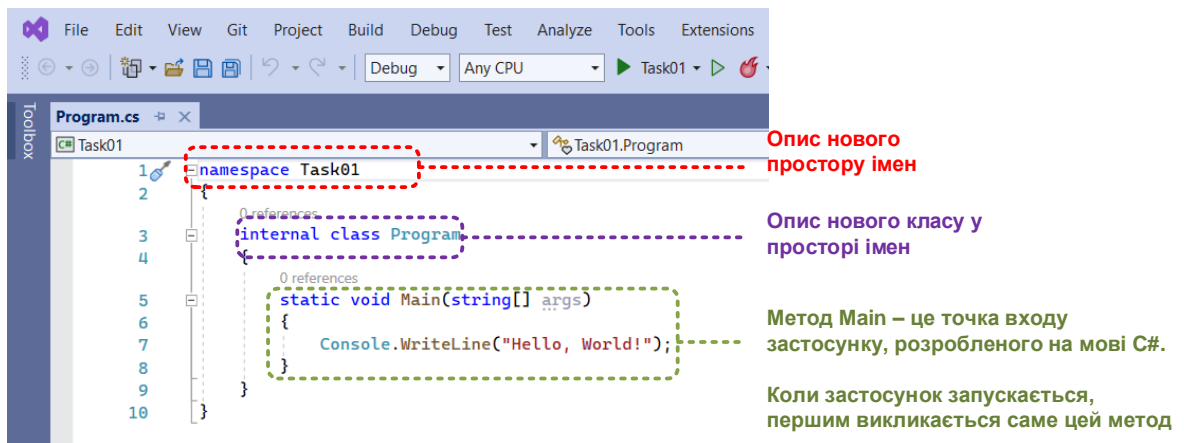
Задайте ім'я проєкту та ім'я рішення (ім'я рішення має містити Ваше прізвище, наприклад Petrenko_ProgrTech_Lab01):



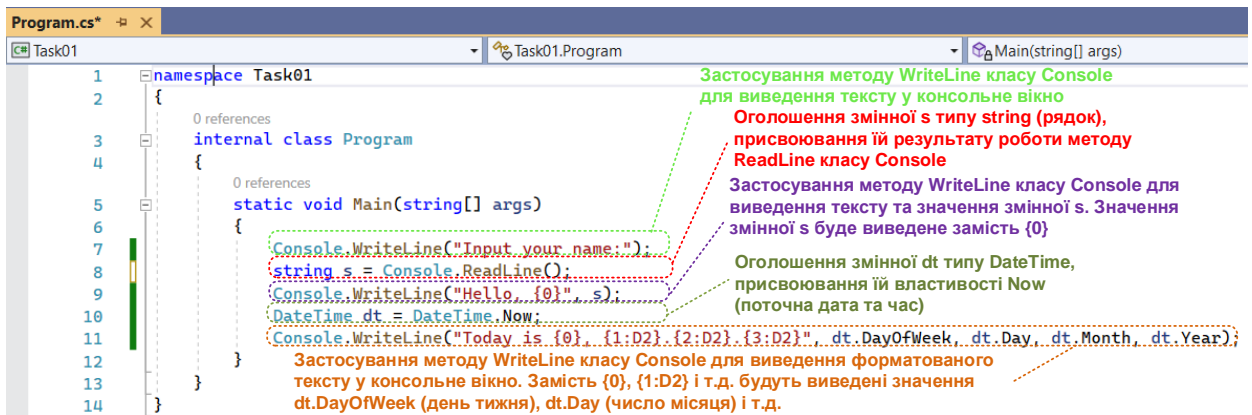
Поставте позначку біля пункту “Do not use top-level statements” для того, щоб було згенеровано шаблон проєкту з методом Main:



Як результат середовищем буде згенеровано такий код:



Наберіть у редакторі такий текст програми:



Запустіть програму на виконання, переконайтеся у відсутності помилок.

Ознайомтеся з можливостями класу Console у MSDN (<https://docs.microsoft.com/en-us/dotnet/api/system.console>), додайте до розробленої раніше програми:

- змінювання кольору тексту;
- змінювання кольору фону;
- змінювання тексту у заголовку консольного вікна;
- змінювання ширини / висоти консольного вікна;
- видачу звукового сигналу.

Ознайомтеся з можливостями структури DateTime у MSDN (<https://docs.microsoft.com/en-us/dotnet/api/system.datetime>), додайте до розробленої раніше програми:

- виведення поточної дати та поточного часу;
- виведення кількості днів, що залишилася до нового року;
- виведення кількості днів, що залишилася до Вашого дня народження;
- виведення кількості днів, що залишилася до найближчого дня ХАІ;
- виведення кількості днів, що залишилася до перемоги.

Завдання 2. Проєктування та розроблення консольного застосунку

Необхідно розробити консольний застосунок, що складається з двох частин – бібліотеки класів, що містить опис типів даних (файл .dll) та програми, що використовує типи даних з цієї бібліотеки (файл .exe).

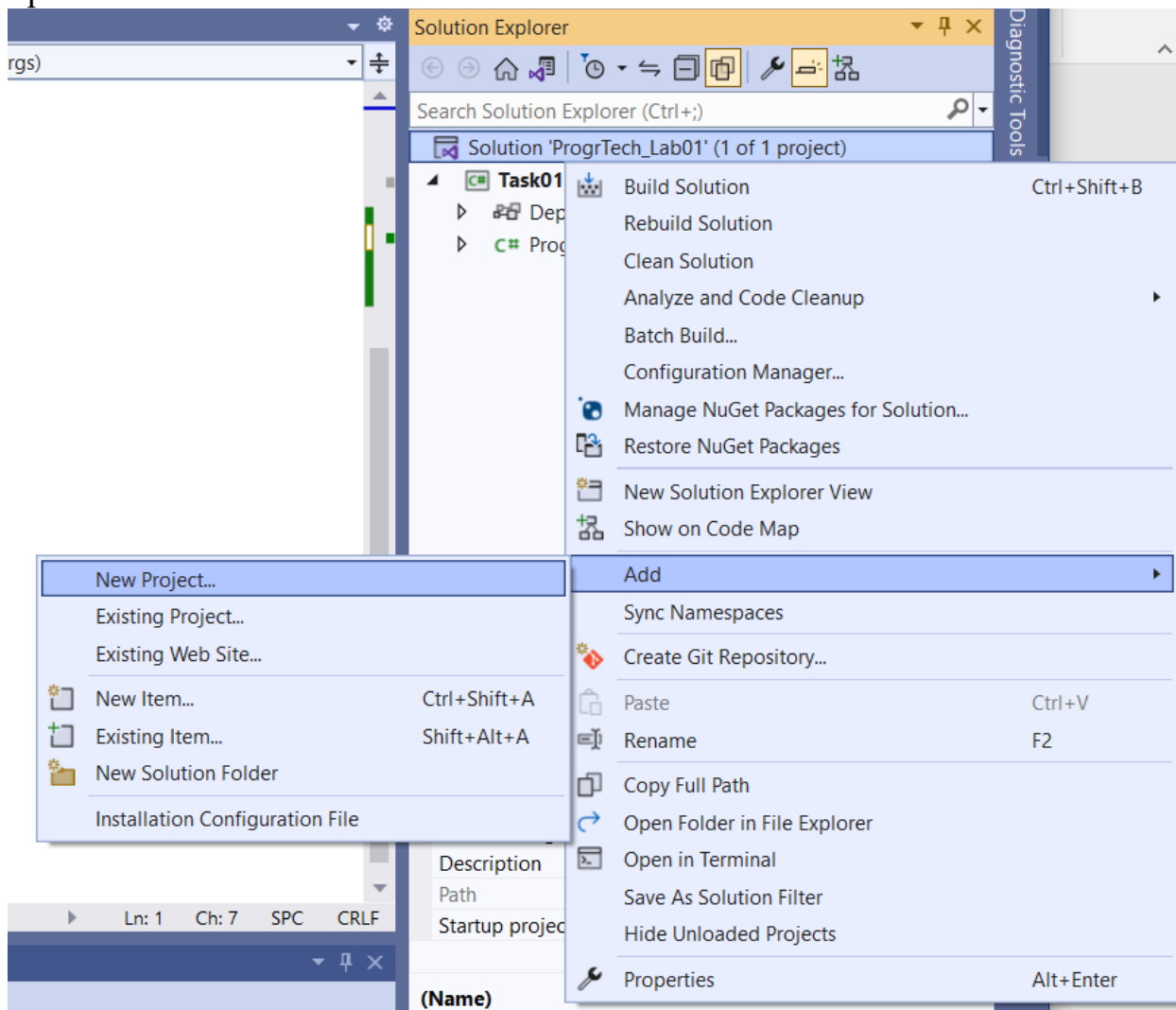
2.1 Створення бібліотеки класів

Теоретичні відомості

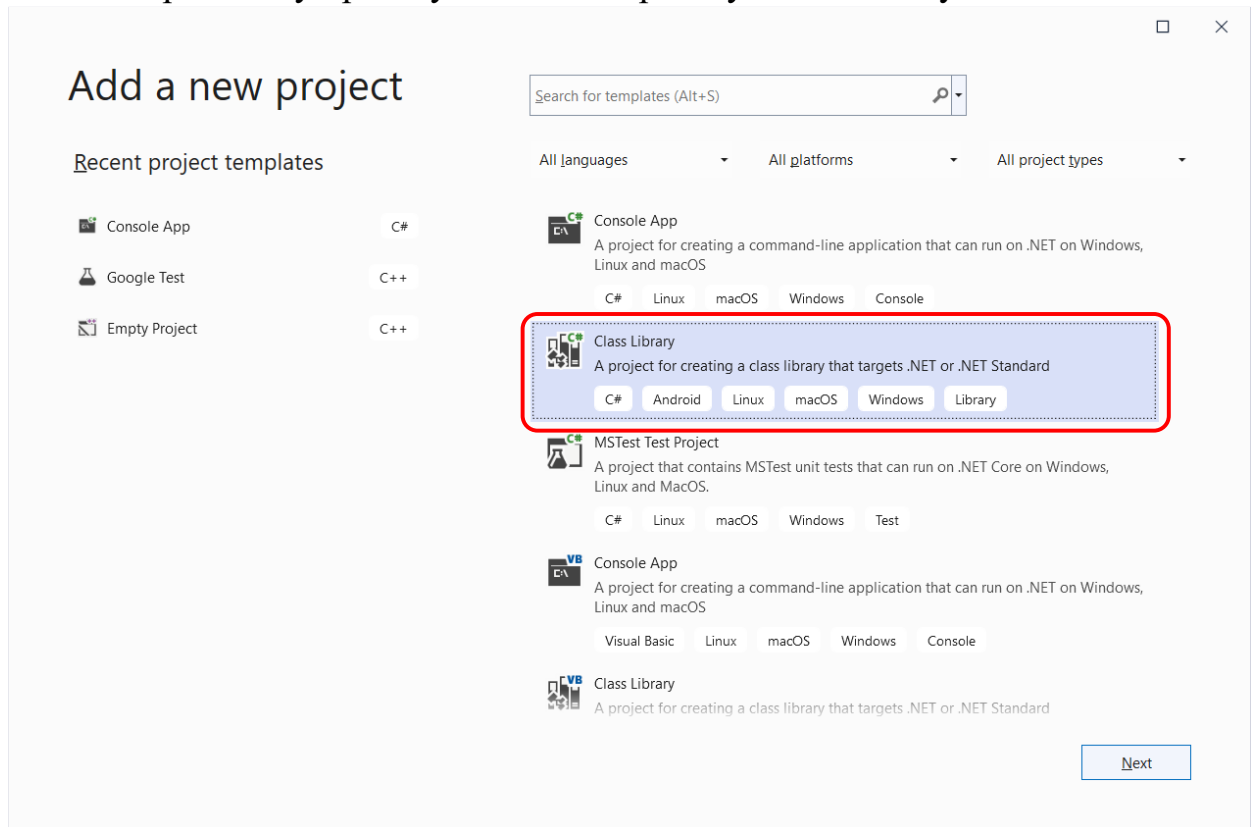
Бібліотека класів містить типи даних (класи і т.д.), що можуть бути використані різними застосунками.

Послідовність виконання

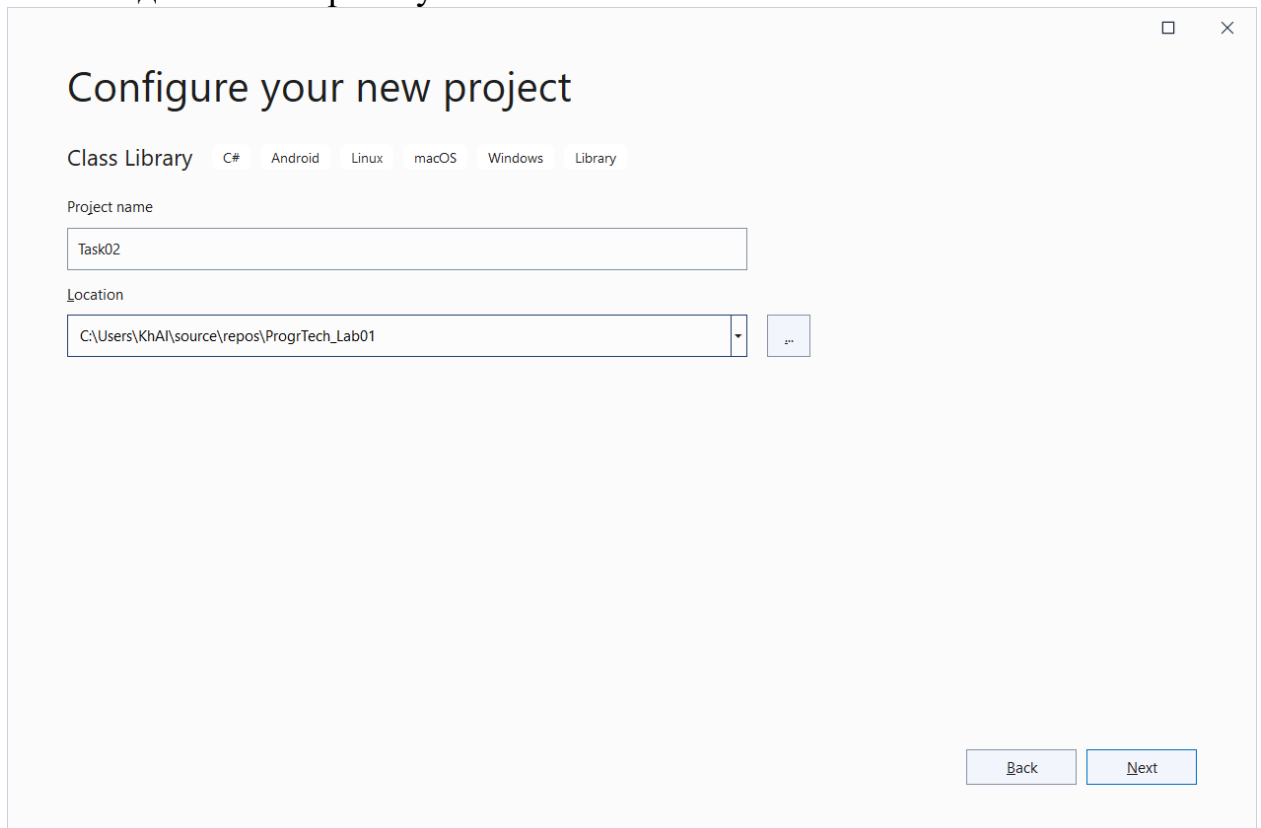
Додайте до рішення ProgrTech_Lab1, створеного у завданні 1, новий проєкт:



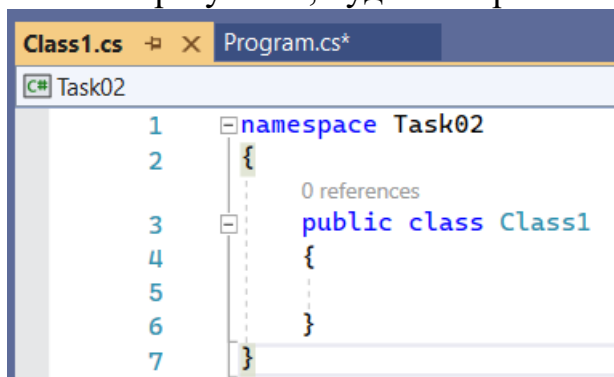
Оберіть мову проєкту C# та тип проєкту Class Library:



Задайте ім'я проєкту:

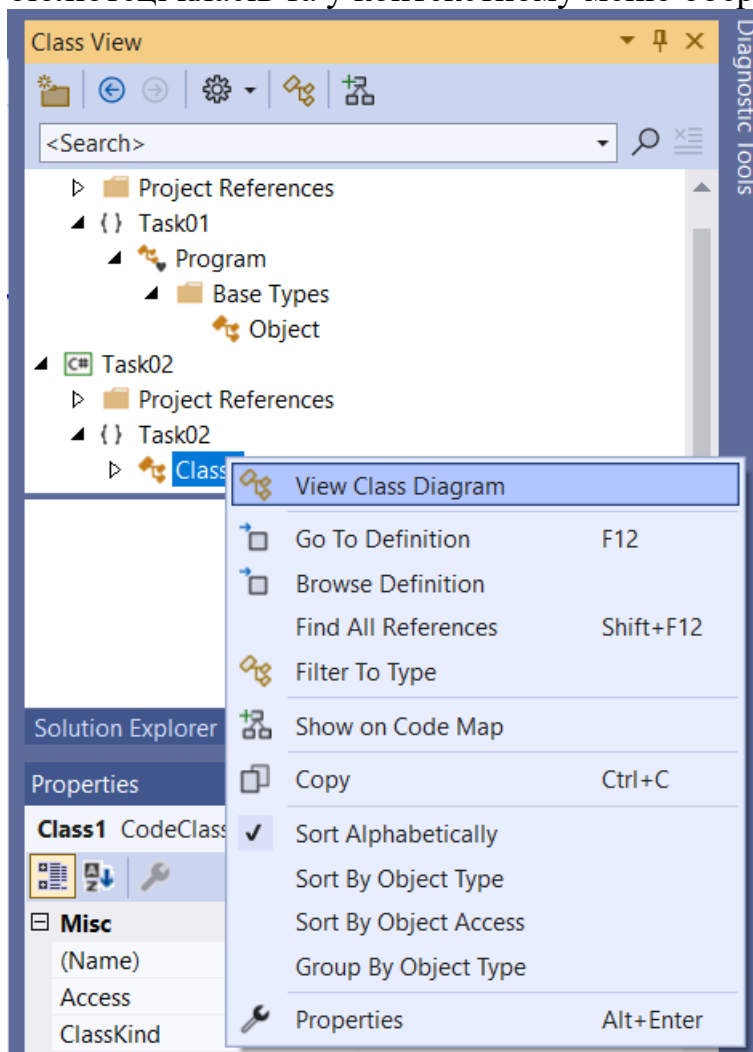


Як результат, буде згенеровано такий програмний код:

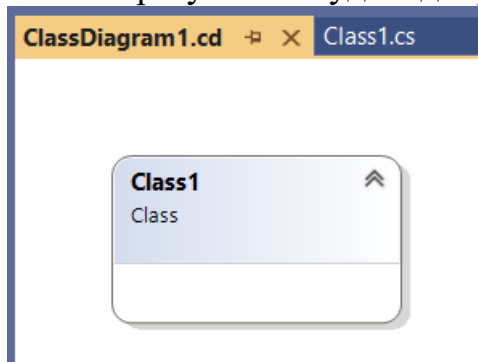


```
1 namespace Task02
2 {
3     0 references
4     public class Class1
5     {
6     }
7 }
```

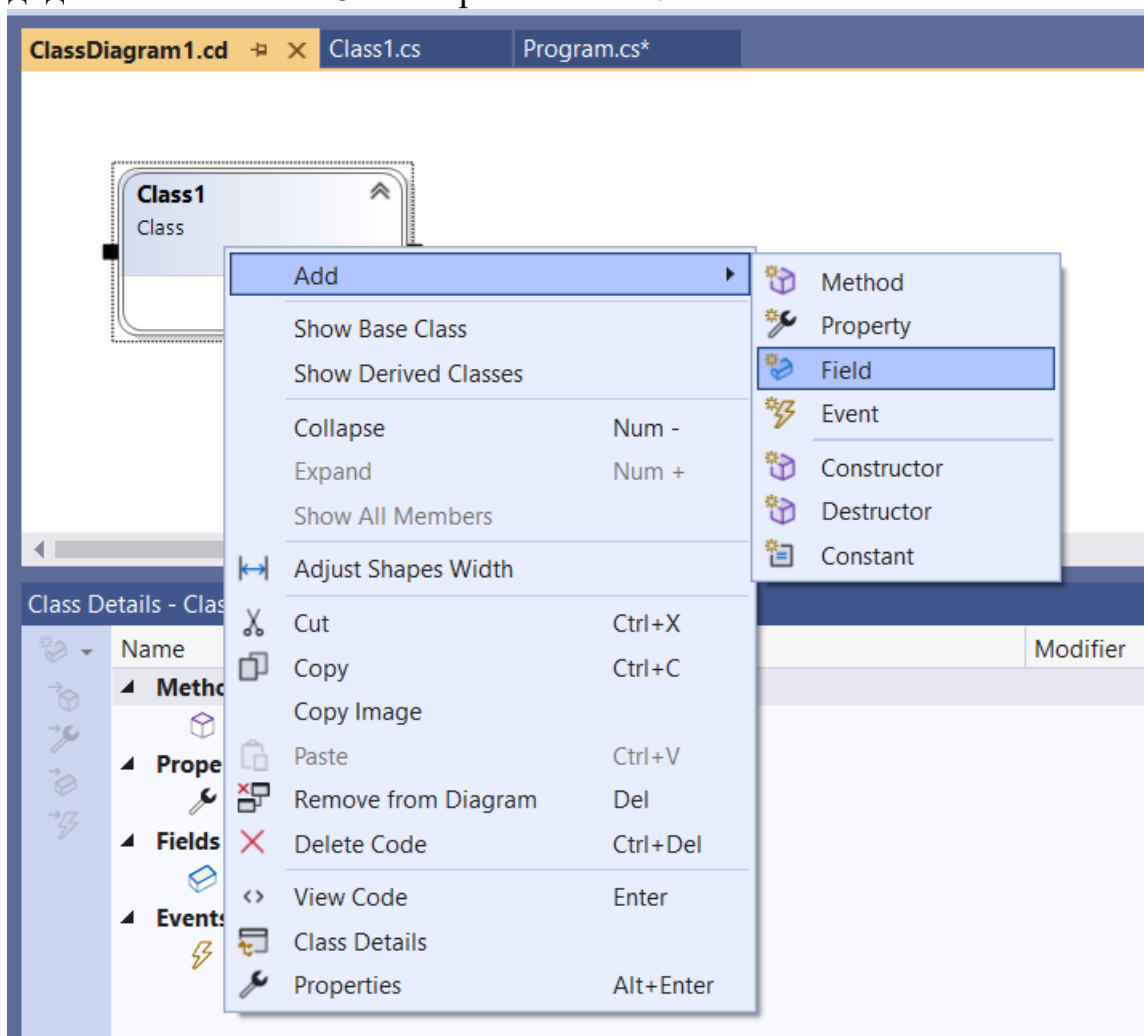
Оберіть пункт меню Class View (View – Class View), оберіть клас у бібліотеці класів та у контекстному меню оберіть пункт View Class Diagram:



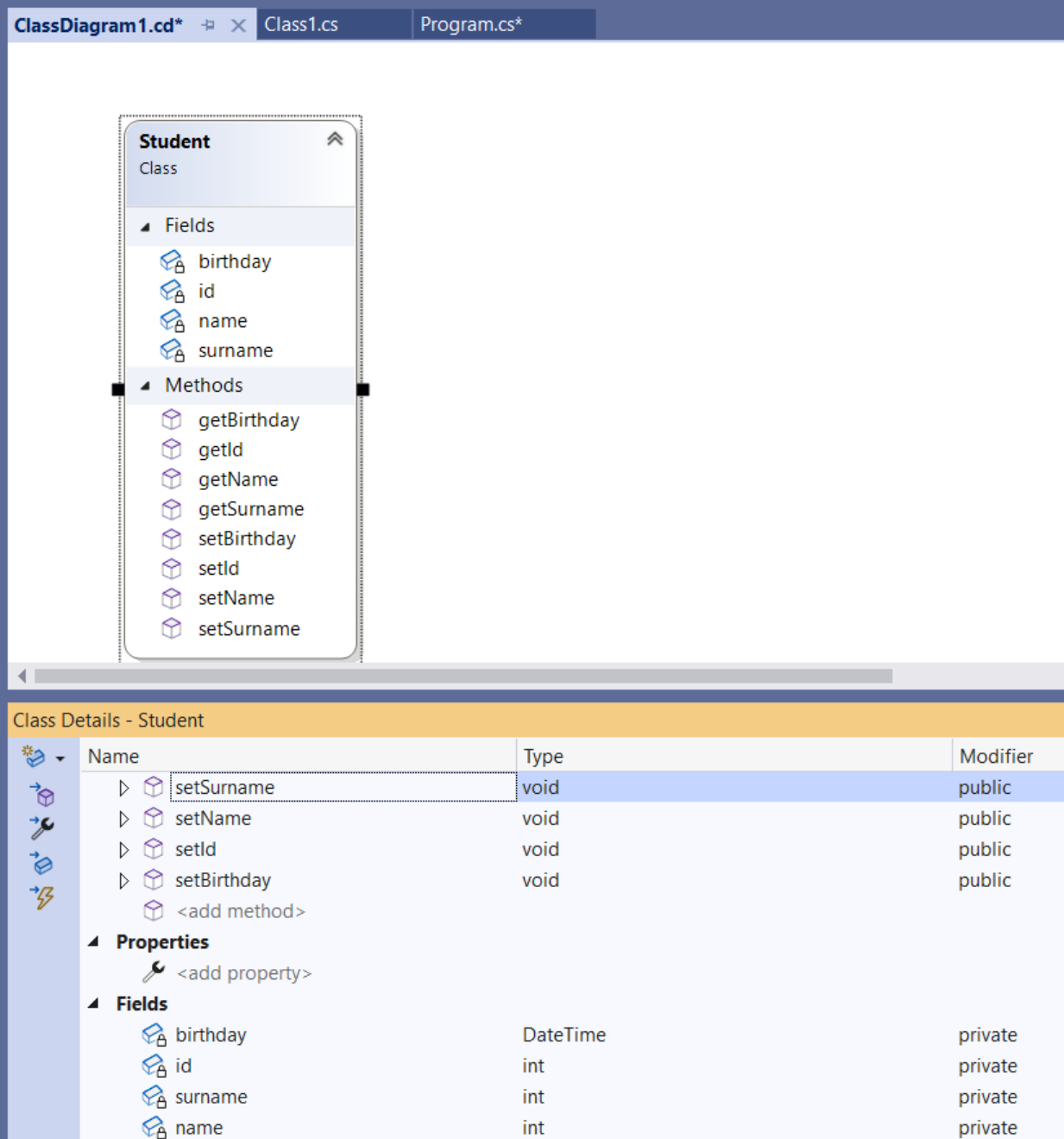
В результаті буде відображено графічний редактор діаграми класів:



Діаграма класів відображає типи даних, їх вміст та зв'язки між ними. Через контекстне меню додайте поля, що характеризують тип даних, який Вам необхідно створити відповідно до варіанту завдання (додаток А) – необхідно додати не менш ніж 5 полів різних типів:



Задайте класу назву відповідно до предметної області, вказаній у варіанті завдання, додайте для кожного поля 2 методи – один для отримання даних поля, другий – для його змінення:

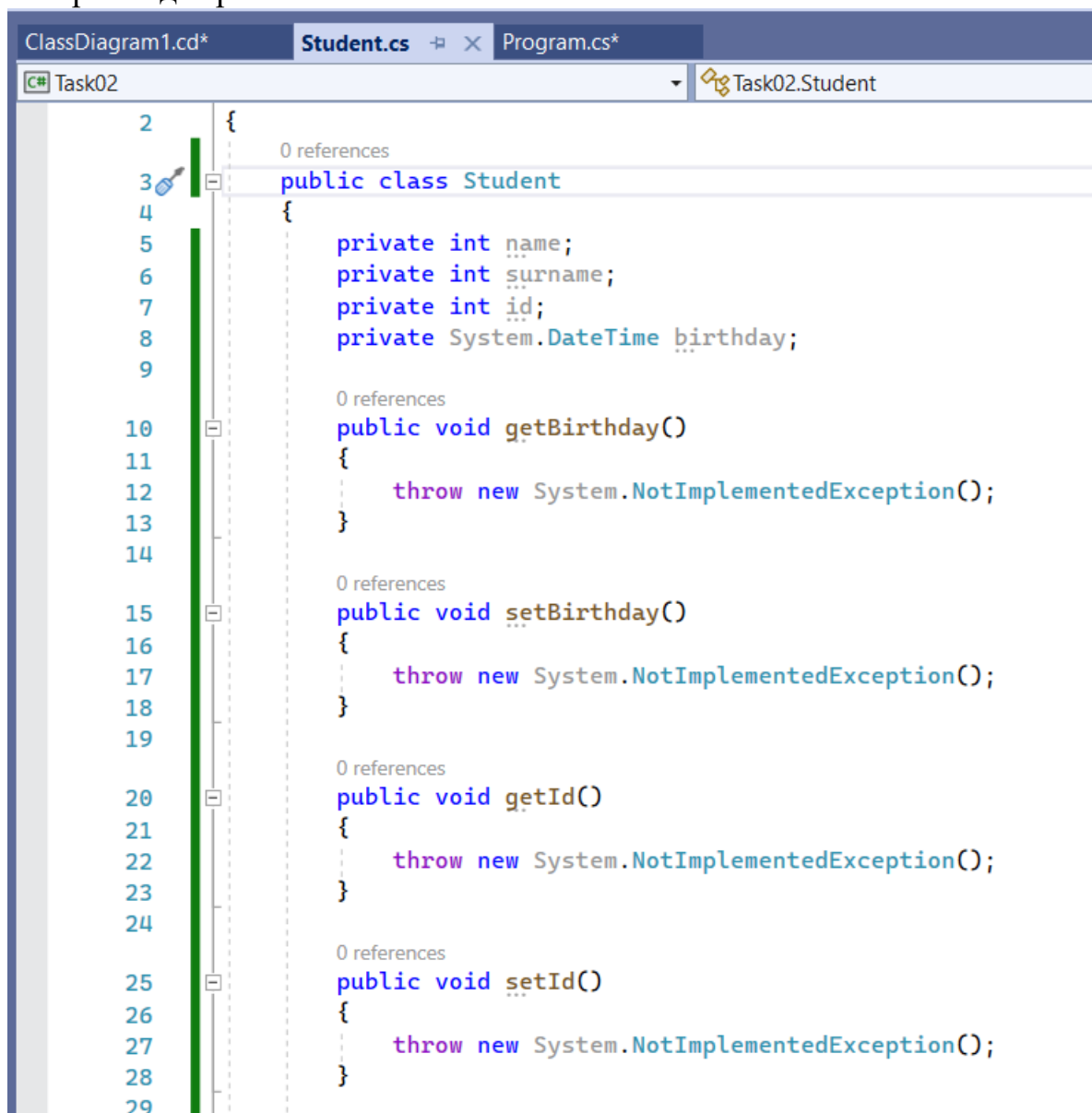


The screenshot displays the Visual Studio IDE with a class diagram for a 'Student' class. The class diagram shows fields (birthday, id, name, surname) and methods (getBirthday, getId, getName, getSurname, setBirthday, setId, setName, setSurname). The 'Class Details - Student' view is open, showing a table of methods and fields.

Name	Type	Modifier
setSurname	void	public
setName	void	public
setId	void	public
setBirthday	void	public
<add method>		
Properties		
<add property>		
Fields		
birthday	DateTime	private
id	int	private
surname	int	private
name	int	private

Зверніть увагу, що за замовчуванням у методів буде модифікатор `public` (можна використовувати за межами класу), а у полів – `private` (поля доступні лише в межах класу).

В результаті середовищем буде згенеровано код, що відповідає створеній діаграмі класів:



```
ClassDiagram1.cd*  Student.cs  Program.cs*
C# Task02  Task02.Student

2  {
3  0 references
4  public class Student
5  {
6      private int name;
7      private int surname;
8      private int id;
9      private System.DateTime birthday;
10
11      0 references
12      public void getBirthday()
13      {
14          throw new System.NotImplementedException();
15      }
16
17      0 references
18      public void setBirthday()
19      {
20          throw new System.NotImplementedException();
21      }
22
23      0 references
24      public void getId()
25      {
26          throw new System.NotImplementedException();
27      }
28
29      0 references
30      public void setId()
31      {
32          throw new System.NotImplementedException();
33      }
34  }
```

Внесіть правки у код (задайте необхідні типи даних, реалізуйте логіку роботи кожного методу, додайте перевірку коректності даних тощо):

```
public class Student
{
    private int name;
    private int surname;
    private int id;
    private System.DateTime birthday;

    0 references
    public DateTime getBirthday()
    {
        return birthday;
    }

    0 references
    public void setBirthday(DateTime newBirthday)
    {
        birthday = newBirthday;
    }

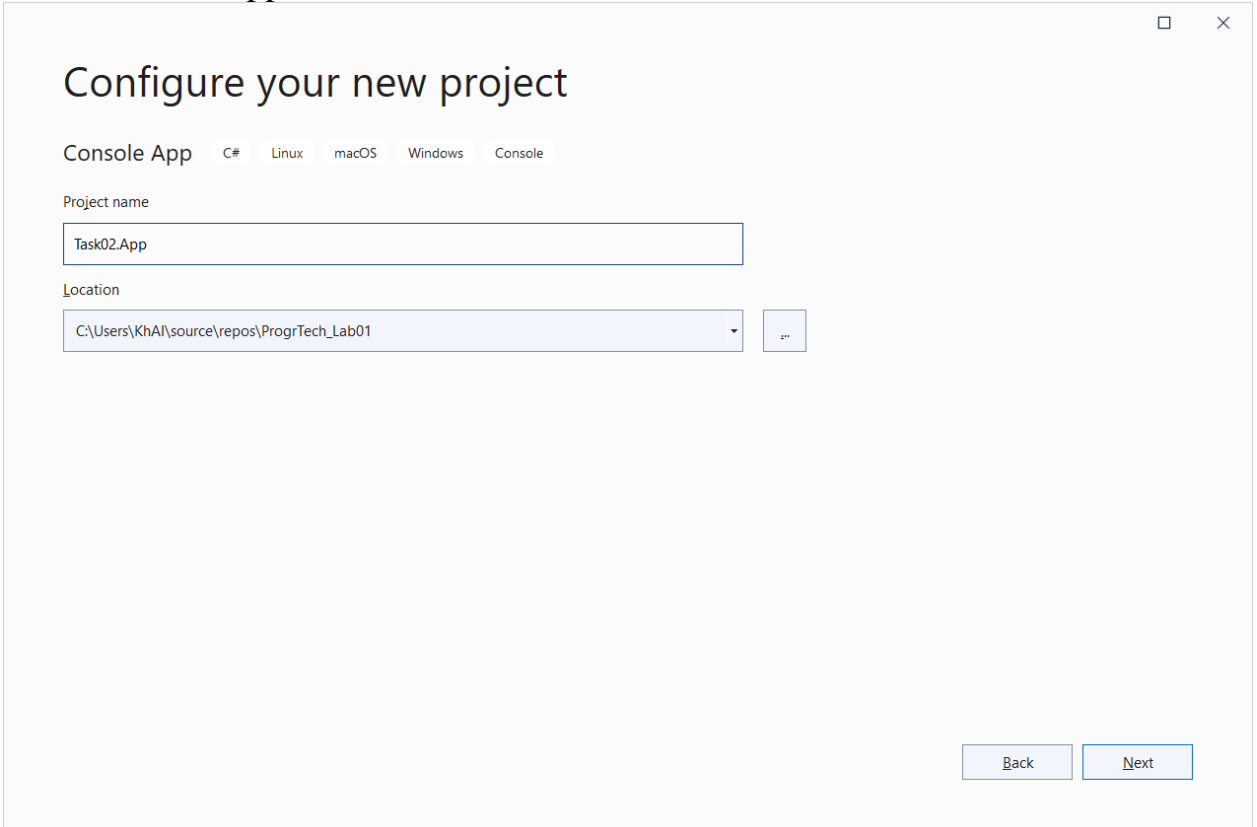
    0 references
    public int getId()
    {
        return id;
    }

    0 references
    public void setId(int newId)
    {
        if (newId > 0) id = newId;
    }
}
```

2.2 Створення консольного застосунку, що використовує класи з бібліотеки класів

Послідовність виконання

Додайте до рішення ProgrTechn_Lab1 проєкт типу Console Application з ім'ям Task02.App:



Configure your new project

Console App C# Linux macOS Windows Console

Project name

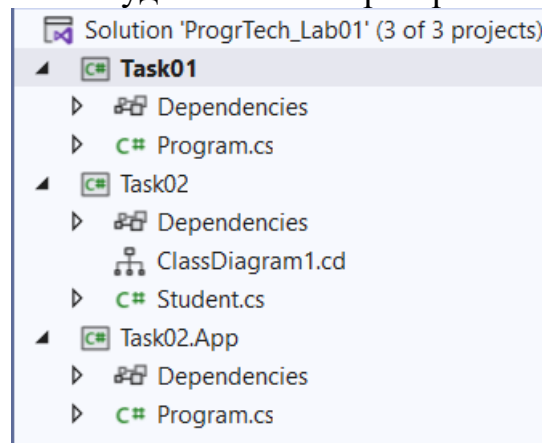
Task02.App

Location

C:\Users\KhAl\source\repos\ProgrTech_Lab01

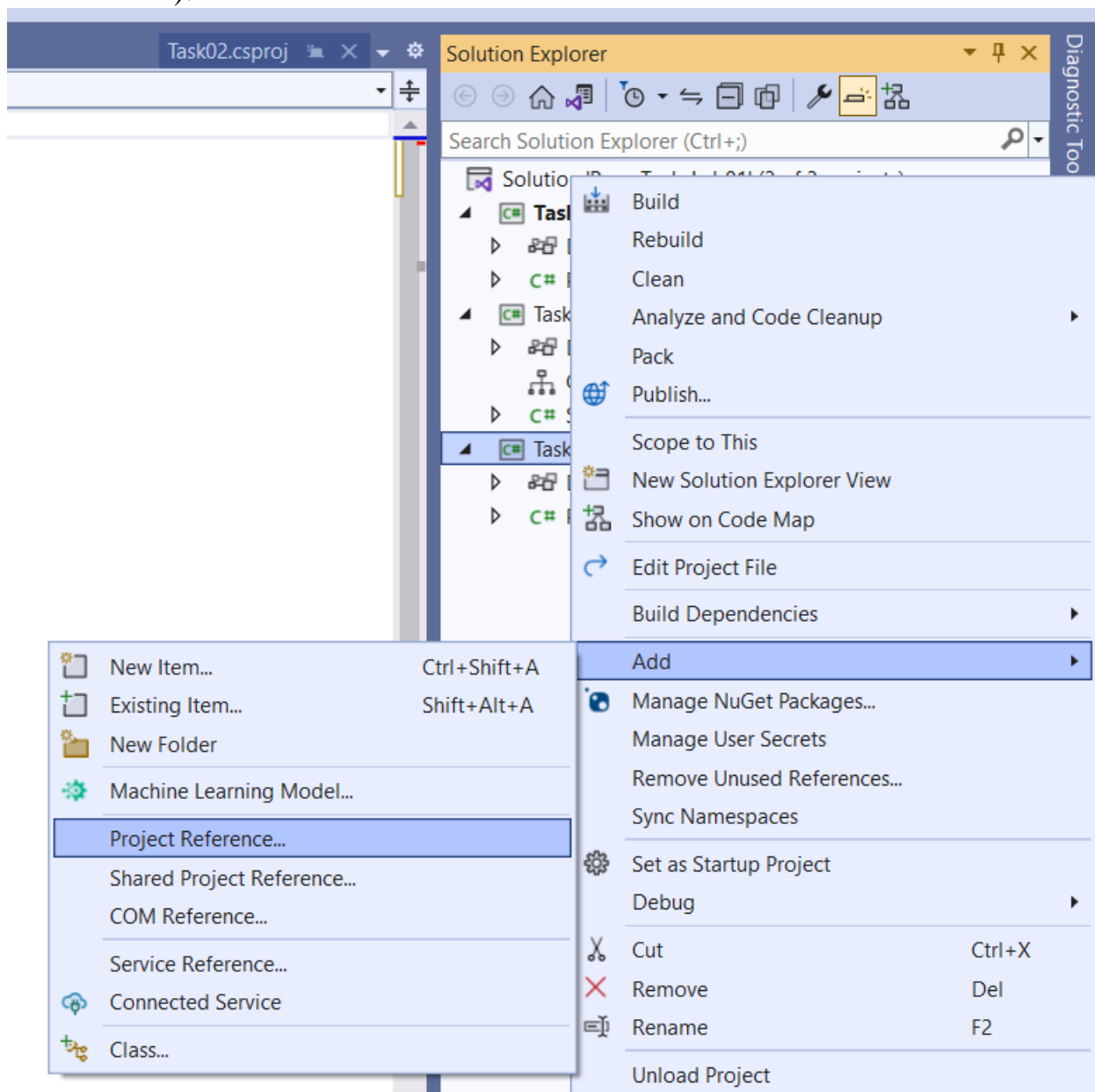
Back Next

В результаті рішення буде включати три проєкти:

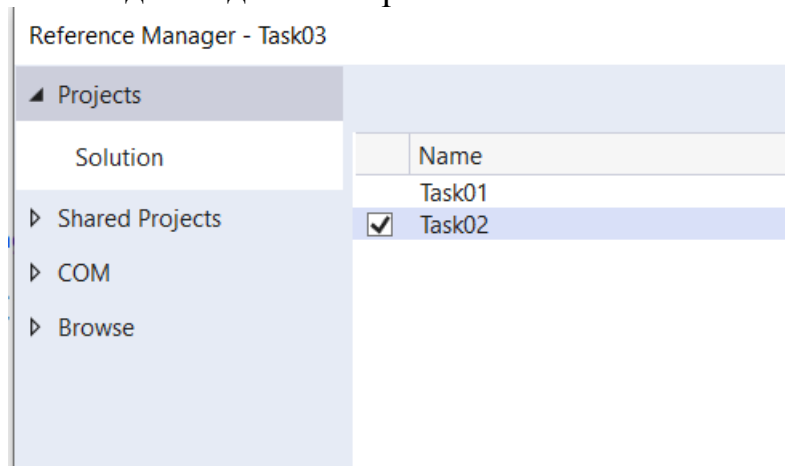


Для того, щоб консольний застосунок міг використовувати наявні у бібліотеці класів типи даних, необхідно додати посилання на неї (у

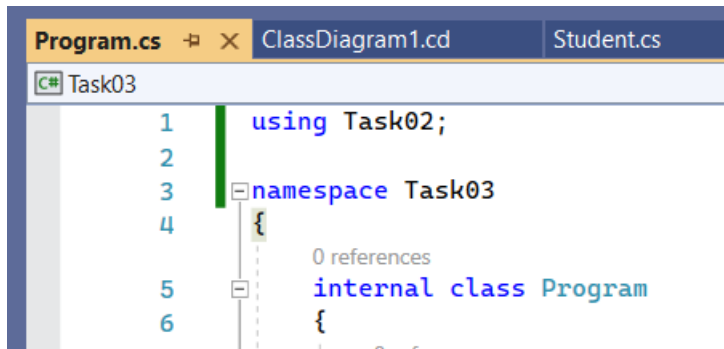
контекстному меню проекту необхідно обрати пункт Add – Project Reference...):



І далі відмітити проект з бібліотекою класів (Task 02):



Крім того, необхідно підключити простір імен бібліотеки класів (Task02 у прикладі):



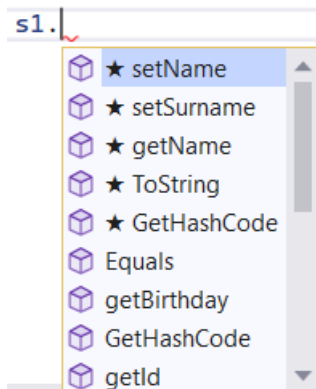
Тепер можна використовувати типи даних, наявні у просторі імен Task02.

Відкрийте метод Main класу Program проєкту Task02.App, опишіть у ньому створення екземпляру (змінної, що має тип даних «клас»), заповнення усіх полів даними та виведення інформації з використанням методів:


```
static void Main(string[] args)
{
    Student s1 = new Student();
    s1.setId(1);
    s1.SetBirthDay(DateTime.Parse("11.01.2004"));
    s1.setName("Petro");
    s1.setSurname("Petrenko");

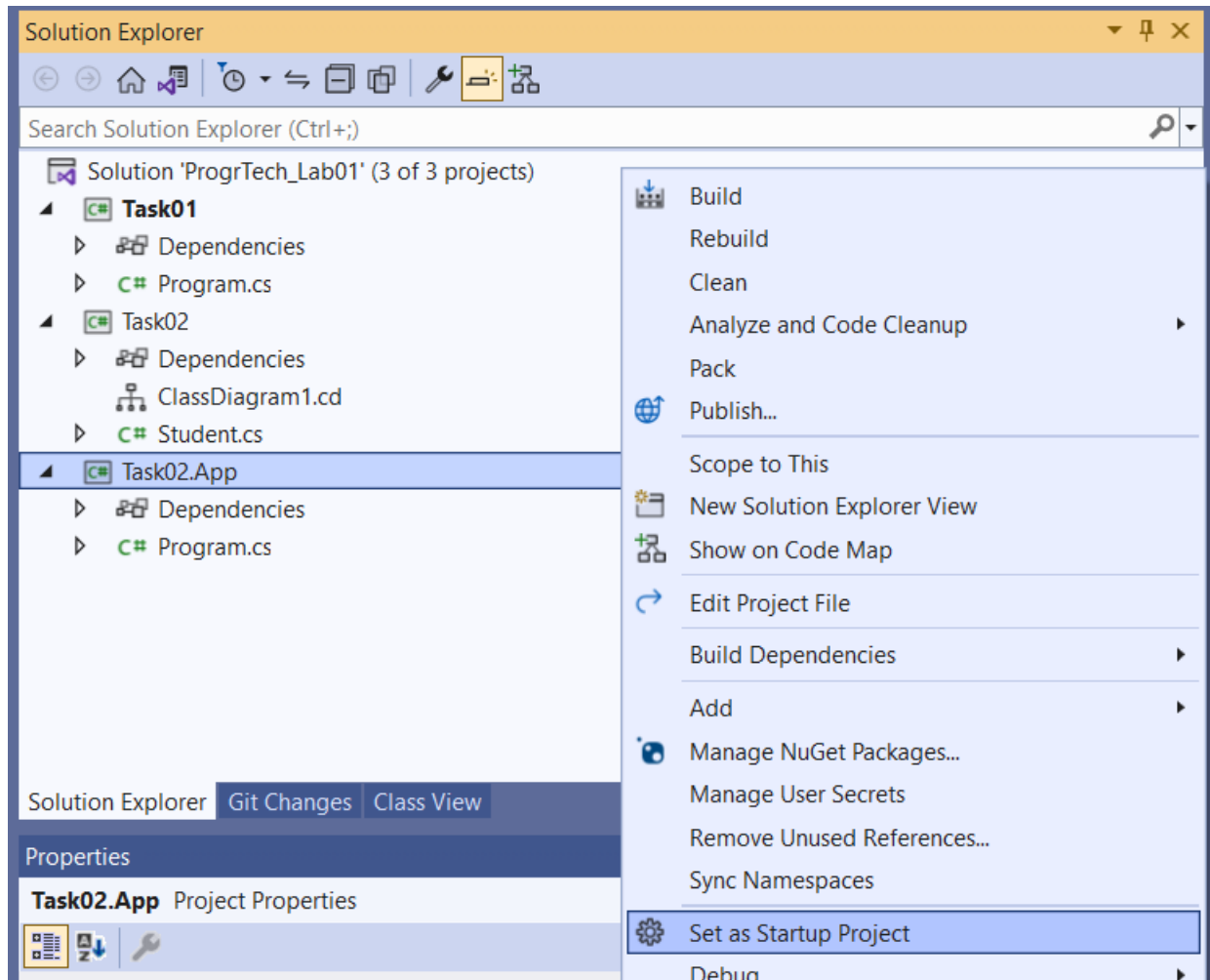
    Console.WriteLine("Student {0} {1} born {2} has id={3}",
        s1.getName(), s1.getSurname(), s1.getBirthDay(), s1.getId());
}
```

Зверніть увагу на варіанти, що пропонує IntelliSense (поля не видні, доступні лише методи):



Запустіть програму на виконання, результати виконання включіть до звіту.

Важливо! Для того, щоб при натисканні кнопки , клавіши F5 або комбінації Ctrl-F5 запускався проєкт Task02.App, необхідно у його контекстному меню обрати пункт «Set as StartUp Project»:



Відкрийте файл з класом, відредагуйте у ньому метод ToText таким чином, щоб він повертав у вигляді рядку інформацію з усіх полів класу, наприклад (зверніть увагу, що метод нічого не виводить у консоль, він лише формує рядок тексту):

```
public string ToText()
{
    return string.Format("{0} {1} (ID {2}), born in {3}, has {4} points", name, surname, ID, birthday.Year, score);
}
```

Оновіть метод Main, створивши у ньому декілька екземплярів, задайте їм різні дані та використайте у всіх екземплярів метод ToText():

```
Console.WriteLine(student1.ToText());
Console.WriteLine(student2.ToText());
Console.WriteLine(student3.ToText());
```

Запустіть програму на виконання, результати включіть до звіту.

Вимоги до звіту

Звіт з лабораторної роботи має включати:

1. титульний аркуш із зазначенням номеру та назви лабораторної роботи;
2. мету роботи;
3. варіант і тексти завдань;
4. діаграму класів (для бібліотеки класів);
5. опис усіх створених класів та членів класів (полів, методів);
6. вихідні тексти програм;
7. результати роботи програм (скріншоти або текст, який виводять програми);
8. тестові набори;
9. висновки (що було зроблено, за допомогою яких засобів, що було вивчено і т.ін.).

Контрольні питання

1. Які мови програмування доступні для розроблення .NET застосунків Microsoft Visual Studio?
2. Що таке рішення у Visual Studio?
3. Що таке проєкт у Visual Studio?
4. Чим відрізняється запуск застосунку клавішею F5 від запуску комбінацією клавіш Ctrl-F5?
5. Що необхідно зробити, щоб при запуску на виконання запускався визначений проєкт?
6. Для чого призначена діаграма класів?
7. Що таке клас? Що таке екземпляр класу?
8. Яке ім'я має бути у метода, що є точкою входу у програму на C#?
9. Вкажіть синтаксис виклику методу, наведіть приклад.
10. За допомогою якого класу / яких методів реалізується виведення у консоль у C#?

Додаток А. Індивідуальні варіанти завдань

1. Програмне забезпечення
2. Побутовий прилад
3. Нерухомість
4. Рахунок у банку
5. Одяг
6. Громадський транспорт
7. Літак
8. Персона
9. Посилка
10. Організація
11. Друковане видання
12. Місто
13. Товар
14. Документ
15. Транспортний засіб
16. Карта пам'яті
17. Держава
18. Тварина
19. Човен
20. Файл
21. Геометрична фігура
22. Мебель
23. Будівля
24. Навчальний заклад
25. Торгова установа