

Лабораторна робота №1

Основи програмування у мові C#. ООП у мові C#. Базові конструкції та типи.

Виконав студент 301-пТК Тараненко Олександр

Завдання

1. Створіть **свою ВЛАСНУ (не як у прикладі)** модель(набір класів) на довільну тематику із 4 або більше класів, серед яких хоча б двоє класів наслідуються від якихось інших. Результуючі класи повинні знаходитись у проєкті {Назва тематики}.Common. Проєкти створювати всередині рішення(.sln) {Назва тематики}.

Кожен клас має мати від 3 властивостей.

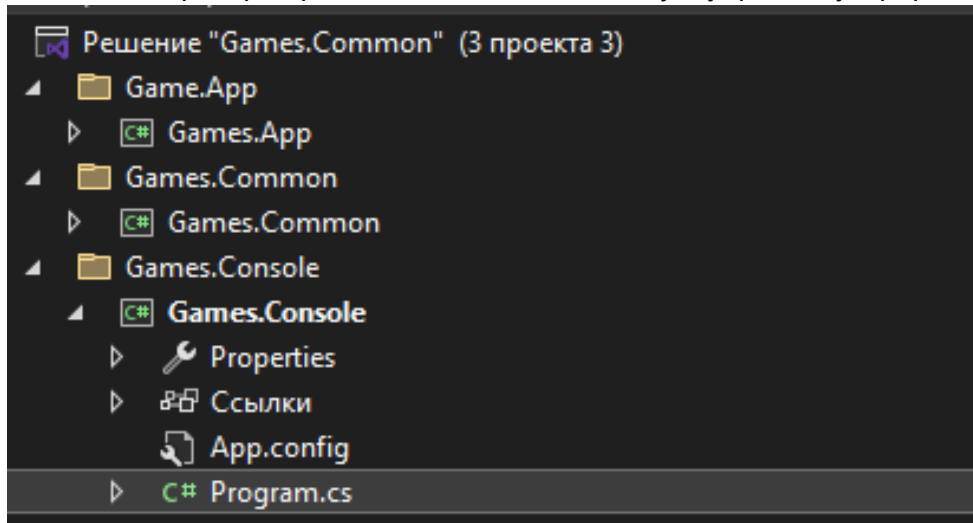
2. Додати у побудовану модель наступні базові конструкції:
 - Методи;
 - Конструктори;
 - Статичні поля і конструктори;
 - Делегати та Події;
 - Статичний методи;
 - Метод розширення.

Застосувати конструкцію 1 чи більше разів, над застосованою конструкцією залишити коментар, приклад:

Реалізувати дженерік CRUD сервіс, який буде зберігати дані у одній із вбудованих колекцій .NET та реалізовуватиме наступний інтерфейс:

3. Створити новий проєкт {Назва тематики}.Console, із консольним застосунком, яким показуватиме як функціонує CRUD сервіс, наприклад створити новий об'єкт CRUD service, додати в нього нові об'єкти, вивести які об'єкти були додані і так далі.

Готова лабораторна робота повинна мати наступну файлову ієрархію:



Результати виконання консольного застосунку.

```
Консоль отладки Microsoft Visual Studio
Вс? ПК ?гри:
Game: Cyberpunk 2077, Price: 59,99, Rating: 7,5
Game: The Witcher 3, Price: 39,99, Rating: 9,5

П?сля знижки на Cyberpunk:
Game: The Witcher 3, Price: 39,99, Rating: 9,5
Game: Cyberpunk 2077, Price: 47,992, Rating: 7,5

П?сля видалення The Witcher 3:
Game: Cyberpunk 2077, Price: 47,992, Rating: 7,5

C:\Users\Monolit\Desktop\NET\NUPP_NET_2025_301pTK_TK_Taranenko_Lab\Lab1\Game.exe (процесс 9828) завершил работу с кодом 0 (0x0).
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно:
```

Додаткове завдання

Додати до CRUD сервісу методи Load(string FilePath) та Save(string FilePath), які будуть зберігати дані із сервісу у серіалізованому вигляді у файлі за шляхом FilePath та завантажувати дані із файлу.

```
Консоль отладки Microsoft Visual Studio

=== В? ПК ?гри ===
Game: Cyberpunk 2077, Price: 59,99, Rating: 7,5
Game: The Witcher 3, Price: 39,99, Rating: 9,5

=== П?сля знижки на Cyberpunk ===
Game: The Witcher 3, Price: 39,99, Rating: 9,5
Game: Cyberpunk 2077, Price: 47,992, Rating: 7,5

=== П?сля видалення The Witcher 3 ===
Game: Cyberpunk 2077, Price: 47,992, Rating: 7,5

Дан? збережено у файл: pcgames.json

=== Дан? п?сля завантаження з файлу ===
Game: Cyberpunk 2077, Price: 47,992, Rating: 7,5

Натисн?ть будь-яку клав?шу для виходу...

C:\Users\Monolit\Desktop\NET\NUPP_NET_2025_301pTK_TK_Taranenko_Lab\Lab1\
ole.exe (процесс 3220) завершил работу с кодом 0 (0x0).
Чтобы автоматически закрывать консоль при остановке отладки, включите пар
томатически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно: _
```

Microsoft.Bcl.AsyncInterfaces.xml	20.08.2025 1:12	Документ XML	31 КБ
pcgames.json	13.09.2025 15:09	JSON File	1 КБ
System.Buffers.dll	19.02.2020 12:05	Расширение при...	21 КБ
System.Buffers.dll	19.02.2020 12:05	Расширение при...	21 КБ

Контрольні запитання

- 1. Що таке класи та структури? Яка різниця між класом та структурою? Яка різниця між value та reference типами?

Клас (class) – це **посилочний тип (reference type)**, який описує об’єкт із властивостями, методами, подіями тощо.

Структура (struct) – це **значущий тип (value type)**, який зазвичай використовується для маленьких об’єктів (наприклад, координати, кольори).

Параметр	Клас (class)	Структура (struct)
Тип	Reference type	Value type
Зберігання	У купі (heap)	На стеку (stack) або в об’єкті
Можна наслідувати	Так	Ні
Конструктор без параметрів	Може бути	За замовчуванням завжди є
Копіювання	Копіюється посилання	Копіюється значення

Value vs Reference:

```
int a = 5;    // value type
int b = a;    // створюється копія
b = 10;
Console.WriteLine(a); // 5

class Person { public string Name; }
Person p1 = new Person() { Name = "Alice" };
Person p2 = p1;    // копіюється посилання
p2.Name = "Bob";
Console.WriteLine(p1.Name); // Bob
```

2. Що таке абстрактний клас? Яка різниця між звичайним та абстрактним класом?

Абстрактний клас (abstract class) – це клас, який не можна інстанціювати безпосередньо, і який може містити абстрактні методи, тобто методи без реалізації.

```
abstract class Animal
{
    public abstract void MakeSound(); // абстрактний метод
}

class Dog : Animal
{
    public override void MakeSound()
    {
        Console.WriteLine("Woof!");
    }
}
```

Різниця:

- Звичайний клас можна створювати (new), абстрактний – ні.
- Абстрактний клас може містити абстрактні методи.
- Звичайний клас має всі методи з реалізацією.

3. Що таке інтерфейс? Яка різниця між абстрактним класом та інтерфейсом?

Інтерфейс (interface) – це контракт, який визначає методи, властивості або події, які клас повинен реалізувати.

```
interface IFlyable
{
    void Fly();
}

class Bird : IFlyable
{
    public void Fly()
    {
        Console.WriteLine("Bird is flying");
    }
}
```

Різниця з абстрактним класом:

- Клас може реалізувати багато інтерфейсів, але успадкувати тільки один клас.
- Інтерфейс не містить стану (полів), абстрактний клас може.
- Абстрактний клас може мати реалізацію методів, інтерфейс – тільки контракт (C# 8+ дозволяє default реалізацію).

4. Що таке наслідування? Які є модифікатори доступу у мові C#?

Наслідування – це механізм, коли один клас отримує властивості та методи іншого.

```
class Vehicle { public int Speed; }  
class Car : Vehicle { public string Model; }
```

Модифікатори доступу в C#:

- `public` – доступно скрізь
- `private` – тільки всередині класу
- `protected` – всередині класу та у похідних класах
- `internal` – всередині збірки
- `protected internal` – поєднання `protected` + `internal`
- `private protected` – доступ лише в межах класу та його похідних у тій же збірці

5. Що таке статичні поля та методи? Яка різниця між звичайними та статичними полями та методами?

Статичні поля та методи (`static`) – належать **класу**, а не конкретному об'єкту.

```
class Counter  
{  
    public static int TotalCount = 0; // статичне поле  
  
    public static void Increment() // статичний метод  
    {  
        TotalCount++;  
    }  
}
```

Різниця:

- Звичайні поля та методи прив'язані до конкретного об'єкта.
- Статичні – до класу, доступ без `new`.

```
Counter.Increment();  
Console.WriteLine(Counter.TotalCount); // 1
```

6. Що таке делегати? Які види делегатів існують?

Делегат (delegate) – це посилання на метод, яке дозволяє викликати методи динамічно.

```
delegate void MyDelegate(string message);

void SayHello(string msg) => Console.WriteLine(msg);

MyDelegate d = SayHello;
d("Привіт!"); // виклик методу через делегат
```

Види делегатів:

- **Одноадресні (Single-cast)** – вказують на один метод
- **Багатоадресні (Multi-cast)** – можуть викликати кілька методів послідовно
- **Вбудовані** – Action, Func, Predicate

```
Action<string> action = SayHello;
Func<int, int, int> sum = (a, b) => a + b;
Predicate<int> isEven = x => x % 2 == 0;
```