

Лабораторна робота №7

Спадкування класів та поліморфізм у мові C#

Мета лабораторної роботи:

1. Вивчити основні концепції відносин між класами C#. Вивчити ставлення агрегування та успадкування.
2. Вивчити прийоми програмування з використанням відносини спадкування класів мовою C#.

Зміст роботи.

1. Теоретичні відомості

1.1 Спадкування та поліморфізм у C#

Явище поліморфізму проявляється у можливості приписати об'єкту не один, а кілька типів. Такі об'єкти називають поліморфними. Поліморфні об'єкти виявляються більш гнучкими у використанні (бо тип визначає поведінку і лімітує використання об'єкта). Наприклад, якщо об'єктам двох різних класів вдасться приписати (додатковий) загальний тип T, їх можна обробити єдиним методом з формальним параметром типу T або зберегти в одному масиві. Основним джерелом поліморфних об'єктів є спадкування класів: об'єкт класу-спадкоємця отримує 2 типи - від свого та батьківського класу.

Спадкування. В описі класу можна вказати батьківський клас, від якого він успадковує. У цьому випадку всі члени батьківського класу (крім private) автоматично додаються до класу-спадкоємця і їх не треба описувати заново. Є можливість перевизначити деякі властивості та методи батька, зберігши їх заголовки. Для цього треба в батьківському класі описати їх virtual, а в спадкоємці - описати наново з модифікатором **override**

[Приклад успадкування класів](#)

2. Завдання на лабораторну роботу

Варіант вибирається за формулою $V = (N \bmod K) + 1$, де N - Ваш номер за академічним журналом групи, K = 10, а mod - операція "залишок від поділу на".

Необхідно виконати 2 завдання. Завдання 1 - загальне всім варіантів. Завдання 2 вибирається згідно з Варіантом.

[Завдання 1 та 2 роботи можна подивитися тут.](#)

3. Звіт

Повинен містити: постановку завдання, програмний код розв'язання, результат роботи написаної програми

Завдання 1 та 2

Завдання 1.(Для всіх варіантів)

Створити базовий клас "Людина". Кожен об'єкт класу повинен містити такі дані:

- ПІБ,
- рік народження,
- підлога,
- статичне поле, що містить кількість створених об'єктів (передбачити статичний блок ініціалізації для цього поля у класі)

Клас повинен містити такі методи:

- Конструктор для ініціалізації інформації;
- Методи доступу (accessors) та модифікуючі методи (mutators) для ВСІХ полів класу;
- Метод WhoAm(), що виводить на консольний висновок назву класу (ідентифікатор, вигаданий Вами для іменування класу «Людина»);
- Метод toString(), що повертає у вигляді рядка інформацію про людину (поточний об'єкт) у довільному вигляді.
- Статичний метод HowMany(), що повертає кількість створених екземплярів класу.

Завдання 2.

Створити клас із зазначеними елементами згідно з варіантом, успадкувавши його від класу «Людина». Крім зазначених у завданні кожного варіанта елементів, необхідно додатково реалізувати:

- статичне поле, що містить кількість створених об'єктів (передбачити статичний блок ініціалізації для цього поля у класі)
- Конструктор для ініціалізації інформації;
- Методи доступу (accessors) та модифікуючі методи (mutators) для ВСІХ полів класу;
- Метод WhoAm(), що виводить на консольний висновок назву класу (ідентифікатор, придуманий Вами для іменування класу згідно з завданням варіанта)
- Метод toString(), що повертає у вигляді рядка інформацію про поточний об'єкт у довільному вигляді.
- Статичний метод HowMany(), що повертає кількість створених екземплярів класу (згідно з варіантом завдання).
- Методи введення та виведення інформації про об'єкт з консолі (можна використовувати акцесори та мутатори)

Варіант 1

Створити похідний клас «Людина» клас Студент, який має додаткові дані:

- рік вступу,
- № залікової книжки,
- кількість дисциплін
- дисципліни, що вивчаються (динамічний масив),
- середній бал.

Клас повинен містити такі методи:

- додавання дисциплін,

- розрахунок середнього балу

Варіант 2

Створити похідний від класу «Людина» клас «Інженер», який має додаткові дані: рік закінчення, ВНЗ,

- спеціальність,
- тип диплома,
- тип навчання,
- перекваліфікація (динамічний масив),
- місце роботи,
- заробітна плата,
- базовий оклад - статична константа. Клас повинен містити такі методи:
- розрахунок заробітної плати (базовий оклад плюс надбавка залежно від кількості перекваліфікацій)
- розрахунок щорічного доходу (приймає кількість відпрацьованих місяців як параметр)
- додавання інформації про перекваліфікацію

Варіант 3.

Створити похідний від класу «Людина» клас «Користувач_бібліотеки», що містить такі дані:

- номер читацького квитка,
- дата видачі,
- перелік прочитаних книг (динамічний масив) — інформація про книгу зберігається у вигляді рядка з вибраним вмістом,
- щомісячний читацький внесок,
- статус користувача.

Клас повинен містити такі методи:

- розрахунок знижки (залежить від кількості прочитаних книг),
- розрахунок щоквартального читацького внеску,
- додавання інформації про прочитані книги.

Варіант 4.

Створити похідний від класу «Людина» клас «Мандрівник», що містить такі дані:

- назва туру,
- дата початку туру,
- дата закінчення туру,
- рівень обслуговування ("****", "*****", "*****", all inclusive", "*****", "*****", all inclusive", ...),
- перелік відвіданих міст та географічних місць (динамічний масив) — інформацію зберігайте у форматі рядка у довільному вигляді,
- ціна відвідування, проживання та екскурсій по кожному об'єкту (динамічний масив – розмірність відповідає розмірності масиву відвіданих об'єктів),

Клас повинен містити такі методи:

- розрахунок вартості туру,
- розрахунок знижки (залежить від кількості відвіданих місць та рівня обслуговування),
- метод IsAllInclusive() – наявність опції «все включено» (визначити з поля рівня обслуговування),
- додавання інформації про відвідані місця.

Варіант 5.

Створити похідний від класу «Людина» клас «Космонавт», що містить такі дані:

- військове звання,
- кількість льотних годин,
- дати тренувальних польотів (динамічний масив),
- дати реальних польотів у космос (☾) (динамічний масив),
- базовий посадовий оклад.

Клас повинен містити такі методи:

- розрахунок зарплати (посадовий оклад плюс надбавки по 1% за кожен тренувальний політ плюс надбавки по 500% за кожен реальний політ),
- IsHerro() – метод, який повертає, чи є космонавт героєм країни (в основному присвоюється за 1 реальний політ у космос, у деяких країнах (Росія) – за 2 реальні польоти),
- додавання інформації про тренувальні та реальні польоти.

Зауваження. При введенні даних зверніть увагу, що в державі Україна було здійснено 1 реальний політ 1 космонавтом.

Варіант 6

Створити похідний від класу «Людина» клас «Програміст», який містить такі дані:

- рівень (junior, middle, senior, team leader),
- кількість відпрацьованих годин за останні 11 місяців (динамічний масив),
- список мов та технологій, якими володіє програміст,
- базова оплата за годину роботи.

Клас повинен містити такі методи:

- розрахунок зарплати за кожен місяць (базова оплата * кількість годин на місяць) - результат - масив зарплат,
- розрахунок відпускних (зарплата за всі місяці + матеріальна допомога у розмірі базова оплата * 150 годин)
- розрахунок квартальної премії (за кожні три ПОВНИХ місяці - у розмірі Базова оплата * Середня кількість годин за три місяці * Коефіцієнт. Коефіцієнт береться або 1, або 1.2 у разі знання C #, або 1.4 у разі знання Java - інформація береться зі списку технологій).

Варіант7.

Створити похідний від класу «Людина» клас «Користувач_бібліотеки», що містить такі дані:

- номер читацького квитка,
- дата видачі,
- перелік прочитаних книг (динамічний масив) — інформація про книгу зберігається у вигляді рядка з вибраним вмістом,
- щомісячний читацький внесок,
- статус користувача.

Клас повинен містити такі методи:

- розрахунок знижки (залежить від кількості прочитаних книг),
- розрахунок щоквартального читацького внеску,
- додавання інформації про прочитані книги.

Варіант8.

Створити похідний від класу «Людина» клас «Мандрівник», що містить такі дані:

- назва туру,

- дата початку туру,
- дата закінчення туру,
- рівень обслуговування ("****", "*****", "*****, all inclusive", "*****", "*****, all inclusive", ...),
- перелік відвіданих міст та географічних місць (динамічний масив) — інформацію зберігайте у форматі рядка у довільному вигляді,
- ціна відвідування, проживання та екскурсій по кожному об'єкту (динамічний масив – розмірність відповідає розмірності масиву відвіданих об'єктів),

Клас повинен містити такі методи:

- розрахунок вартості туру,
- розрахунок знижки (залежить від кількості відвіданих місць та рівня обслуговування),
- метод IsAllInclusive() – наявність опції «все включено» (визначити з поля рівня обслуговування),
- додавання інформації про відвідані місця.

Варіант9.

Створити похідний від класу «Людина» клас «Космонавт», що містить такі дані:

- військове звання,
- кількість льотних годин,
- дати тренувальних польотів (динамічний масив),
- дати реальних польотів у космос (☾) (динамічний масив),
- базовий посадовий оклад.

Клас повинен містити такі методи:

- розрахунок зарплати (посадовий оклад плюс надбавки по 1% за кожен тренувальний політ плюс надбавки по 500% за кожен реальний політ),
- IsHero() – метод, який повертає, чи є космонавт героєм країни (в основному присвоюється за 1 реальний політ у космос, у деяких країнах (Росія) – за 2 реальні польоти),
- додавання інформації про тренувальні та реальні польоти.

Зауваження. При введенні даних зверніть увагу, що в державі Україна було здійснено 1 реальний політ 1 космонавтом.

Варіант10

Створити похідний від класу «Людина» клас «Програміст», який має додаткові дані: рік закінчення, ВНЗ,

- спеціальність,
- тип диплома,
- тип навчання,

- перекваліфікація (динамічний масив),
- місце роботи,
- заробітна плата,
- базовий оклад - статична константа. Клас повинен містити такі методи:
- розрахунок заробітної плати (базовий оклад плюс надбавка залежно від кількості перекваліфікацій)
- розрахунок щорічного доходу (приймає кількість відпрацьованих місяців як параметр)
- додавання інформації про перекваліфікацію

Приклад успадкування класів

```
using System;
namespace InheritanceGuide
{
    /* Classes може вносити з іншого класу. Class B inherit від class A (called basic):
    * public class B: A
    * {
    * //Additional members, New definitions for some (virtual) methods from A
    * }
    * The new class – the derived class B – then gains all the non-private data and behavior
    * of the base class in addition to any other data or behaviors it defines for itself.
    * The new class then has 2 effective types: the type of the new class and the type of
    * the class it inherits (polymorphism).
    */
    public class AClass
    {
        public int n = 10;
        private string mykey = "unknown";

        public string Key
        {
            get {return mykey; }
            set {mykey = value; }
        }
        public void Meth1()
        {
            for (int i = 0; i <= n; i++)
            {
                Console.WriteLine("{0}", i);
            };
            Console.WriteLine();
        }
        public virtual void Meth2() // "virtual" - цей метод може бути overridden.
                                   // see also: "abstract" = "virtual"+(empty implementation here; must be overridden)
        {
            Console.WriteLine("Original Meth2() says: ");
            Meth1();
        }
    }
    public class BClass : AClass
    {
        public void Meth3()
        {
            Console.WriteLine("Additional Meth3() from BClass");
        }
    }
}
```

```

}

// Забарвлення modifier є необхідною для розширення або змінити abstract або virtual
// implementation of an inherited method, property, indexer, or event.

public override void Meth2() //new definition that overrides one from AClass
{
    Console.WriteLine("\n--- overridden Meth2() ---");
    base.Meth2();
    int summ = 0;
    for (int i = 0; i <= n; i++)
    {
        summ += i;
    };
    Console.WriteLine("summ = {0}", summ);
    Console.WriteLine("--- overridden Meth2() done---\n");
}
/* There is another way to replace an inherited method -- "нове" keyword.
 * It is the straightforward (static) shadowing of the base method.
 */
}

```

class Program

```

{
    static void Main()
    {
        /*****/
        AClass a = new AClass();
        Console.WriteLine("\na : {0}", a.ToString());
        Console.WriteLine(a.Key);
        a.Key = "privet";
        Console.WriteLine(a.Key);
        an = 5;
        a.Meth1();
        a.Meth2();
        Console.ReadKey();

        /*****/
        BClass b = New BClass ();
        Console.WriteLine("\nb : {0}", b.ToString());
        Console.WriteLine(b.Key);
        b.Key = "bol'shoi privet";
        Console.WriteLine(b.Key);
        bn = 5;
        b.Meth1();
        b.Meth2();
        b.Meth3();
    }
}

```

```
Console.ReadKey();

/*****
 * Declare arr as AClass array, and assign AClass and BClass values.
 * The right variants of Meth2() will be called.
 */
Console.WriteLine("\n=====");
AClass[] arr = new AClass[5];
arr[0] = новый BClass();
arr[1] = новый AClass();
arr[2] = новый AClass();
arr[3] = новый AClass();
arr[4] = новый BClass();
for (int i = 0; i < 5; i++)
{arr[i].Meth2(); }

}

}
```