# **C# Polymorphism**

**Поліморфізм та методи перевизначення**

**(Polymorphism and Overriding Methods)**

Поліморфізм означає *"багато форм"*, і він виникає, коли у нас є багато класів, які пов'язані один з одним за допомогою успадкування.

Успадкування дозволяє нам успадковувати поля і методи з іншого класу. Поліморфізм використовує ці методи для виконання різних завдань. Це дозволяє нам виконувати одну і ту ж дію різними способами.

Наприклад, уявіть собі базовий клас ***Animal***, який має метод ***animalSound()***. Похідними класами Тварин можуть бути Свині, Коти, Собаки, Птахи - і вони також мають власну реалізацію звуку тварини (свиня хрюкає, кіт нявкає і т.д.):

class Animal // Base class (parent)

{

public void animalSound()

{

Console.WriteLine("The animal makes a sound");

}

}

class Pig : Animal // Derived class (child)

{

public void animalSound()

{

Console.WriteLine("The pig says: wee wee");

}

}

class Dog : Animal // Derived class (child)

{

public void animalSound()

{

Console.WriteLine("The dog says: bow wow");

}

}

Тепер ми можемо створити об'єкти ***Pig*** та ***Dog*** і викликати метод ***animalSound()*** для них обох:

class Animal // Base class (parent)

{

public void animalSound()

{

Console.WriteLine("The animal makes a sound");

}

}

class Pig : Animal // Derived class (child)

{

public void animalSound()

{

Console.WriteLine("The pig says: wee wee");

}

}

class Dog : Animal // Derived class (child)

{

public void animalSound()

{

Console.WriteLine("The dog says: bow wow");

}

}

class Program

{

static void Main(string[] args)

{

Animal myAnimal = new Animal(); // Create a Animal object

Animal myPig = new Pig(); // Create a Pig object

Animal myDog = new Dog(); // Create a Dog object

myAnimal.animalSound();

myPig.animalSound();

myDog.animalSound();

}

}

The output will be:

The animal makes a sound

The animal makes a sound

The animal makes a sound

**Не той результат, на який я розраховував**

У наведеному вище прикладі ви, ймовірно, отримали не той результат, який очікували. Це тому, що метод базового класу перевизначає метод похідного класу, якщо вони мають однакове ім'я.

Однак, C# надає можливість перевизначити метод базового класу, додавши ключове слово **virtual** до методу всередині базового класу, а також використовуючи ключове слово **override** для кожного методу похідного класу:

class Animal // Base class (parent)

{

public **virtual** void animalSound()

{

Console.WriteLine("The animal makes a sound");

}

}

class Pig : Animal // Derived class (child)

{

public **override** void animalSound()

{

Console.WriteLine("The pig says: wee wee");

}

}

class Dog : Animal // Derived class (child)

{

public **override** void animalSound()

{

Console.WriteLine("The dog says: bow wow");

}

}

class Program

{

static void Main(string[] args)

{

Animal myAnimal = new Animal(); // Create a Animal object

Animal myPig = new Pig(); // Create a Pig object

Animal myDog = new Dog(); // Create a Dog object

myAnimal.animalSound();

myPig.animalSound();

myDog.animalSound();

}

}

The output will be:

The animal makes a sound

The pig says: wee wee

The dog says: bow wow