**C# Abstraction**

**Абстракція C#**

**Абстрактні класи та методи**

***Абстрагування даних*** - це процес приховування певних деталей і показу користувачеві лише найважливішої інформації.

Абстракція може бути досягнута за допомогою абстрактних класів або інтерфейсів.

Ключове слово **abstract** використовується для класів і методів:

* **Абстрактний клас**: це обмежений клас, який не можна використовувати для створення об'єктів (щоб отримати доступ до нього, він повинен бути успадкований від іншого класу).
* **Абстрактний метод**: може бути використаний тільки в абстрактному класі і не має тіла. Тіло надається похідним класом (успадкованим від нього).

Абстрактний клас може мати як абстрактні, так і звичайні методи:

abstract class Animal

{

public abstract void animalSound();

public void sleep()

{

Console.WriteLine("Zzz");

}

}

З вищенаведеного прикладу неможливо створити об'єкт класу ***Animal***:

Animal myObj = new Animal(); // Will generate an error (Cannot create an instance of the abstract class or interface 'Animal')

Щоб отримати доступ до абстрактного класу, він повинен бути успадкований від іншого класу.

// Abstract class

abstract class Animal

{

// Abstract method (does not have a body)

public abstract void animalSound();

// Regular method

public void sleep()

{

Console.WriteLine("Zzz");

}

}

// Derived class (inherit from Animal)

class Pig : Animal

{

public override void animalSound()

{

// The body of animalSound() is provided here

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

}

}

class Program

{

static void Main(string[] args)

{

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

myPig.animalSound(); // Call the abstract method

myPig.sleep(); // Call the regular method

}

}

**Навіщо і коли використовувати абстрактні класи та методи?**

Для досягнення безпеки - приховати певні деталі і показати лише важливі деталі об'єкта.

Примітка: Абстракція також може бути досягнута за допомогою інтерфейсів.