



INSTITUTO
KHIPU

Semestre III

Sesión 16

CARRERA PROFESIONAL

DESARROLLO DE SISTEMAS DE INFORMACIÓN

**HERRAMIENTAS DE
PROGRAMACION C#**

Tema

**HERENCIA DE CLASES – Palabras clave
Protected, base, override, virtual**

HERENCIA DE CLASES – Palabras clave Protected, base, override, virtual

Palabra Clave protected

La palabra clave **protected** es un modificador de acceso que permite que un miembro de una clase (propiedad o método) sea accesible desde:

- La propia clase.
- Las clases derivadas.

Es más restrictivo que public, pero más accesible que private. Los miembros protected no son accesibles fuera de la clase o de las clases derivadas.

Ejemplo de protected

```
class Animal
{
    protected string Especie { get; set; }

    protected void Descripcion()
    {
        Console.WriteLine($"Este animal es un {Especie}.");
    }
}

class Perro : Animal
{
    public void MostrarDescripcion()
    {
        // Acceso a Los miembros protected de la clase base
        Especie = "Perro";
        Descripcion();
    }
}
```

Palabra Clave virtual

La palabra clave virtual se usa para declarar un método o propiedad en la clase base que puede ser sobrescrito en una clase derivada. Esto es útil cuando quieres que las clases derivadas puedan personalizar o modificar el comportamiento de un método específico.

Ejemplo de virtual

```
class Animal
{
    public virtual void HacerSonido()
    {
        Console.WriteLine("El animal hace un sonido.");
    }
}

class Perro : Animal
{
    // Este método sobrescribe el comportamiento del método en la clase
    public override void HacerSonido()
    {
        Console.WriteLine("El perro ladra.");
    }
}
```

Palabra Clave override

override se usa para sobrescribir un método o propiedad virtual de la clase base. Este permite redefinir el comportamiento de un método heredado en la clase derivada.

Ejemplo de override

```
class Gato : Animal
{
    // Este método sobrescribe el comportamiento del método en la clase b
    public override void HacerSonido()
    {
        Console.WriteLine("El gato maúlla.");
    }
}
```

Palabra Clave base

La palabra clave base se utiliza para acceder a miembros de la clase base desde una clase derivada. Esto es útil cuando una clase derivada quiere reutilizar el comportamiento de la clase base, pero agregar más funcionalidad.

Ejemplo de base

```
class Animal
{
    public string Nombre { get; set; }

    public virtual void Descripcion()
    {
        Console.WriteLine($"{Nombre} es un animal.");
    }
}

class Perro : Animal
{
    public override void Descripcion()
    {
        // Llamada al método Descripcion() de la clase base
        base.Descripcion();
        Console.WriteLine($"{Nombre} es un perro.");
    }
}
```

Ejemplo Completo usando protected, virtual, override y base

```
class Animal
{
    // Miembro protegido que es accesible en las clases derivadas
    protected string Especie { get; set; }

    // Método virtual que puede ser sobrescrito
    public virtual void HacerSonido()
    {
        Console.WriteLine("El animal hace un sonido.");
    }

    public Animal(string especie)
    {
        Especie = especie;
    }
}
```

```
class Perro : Animal
{
    // Propiedad adicional para la clase derivada
    public string Raza { get; set; }

    // Constructor de la clase derivada que utiliza base para inicializar la clase base
    public Perro(string especie, string raza) : base(especie)
    {
        Raza = raza;
    }

    // Sobrescribir el método virtual de la clase base
    public override void HacerSonido()
    {
        base.HacerSonido(); // Llamada al método original de la clase base
        Console.WriteLine($"El {Especie} de raza {Raza} ladra.");
    }
}
```

```
class Programa
{
    static void Main(string[] args)
    {
        Perro miPerro = new Perro("Canino", "Labrador");
        miPerro.HacerSonido();
    }
}
```

Explicación:

1. protected string Especie es una propiedad que solo puede ser accedida por la clase base Animal y sus derivadas.
2. El método HacerSonido() en la clase base es virtual, lo que permite que las clases derivadas lo sobrescriban.
3. En la clase Perro, se utiliza override para redefinir el método HacerSonido().
4. Se utiliza base dentro de Perro para llamar al método HacerSonido() de la clase base Animal.

Conclusión

Las palabras clave protected, virtual, override y base son fundamentales en la herencia de clases en C#, ya que permiten controlar el acceso a los miembros, definir comportamientos personalizables en clases derivadas y reutilizar el código de la clase base.



INSTITUTO
KHIPU