INSTITUTO KHIPU

Semestre III

Sesión 17

CARRERA PROFESIONAL

DESARROLLO DE SISTEMAS DE INFORMACIÓN

HERRAMIENTAS DE PROGRAMACION C#

Tema

CLASES ABSTRACT E INTERFACES - Clases metodo abstract, Interfaces

CLASES ABSTRACT E INTERFACES - Clases metodo abstract, Interfaces

En C#, tanto las clases abstractas como las interfaces se utilizan para definir contratos de comportamiento que otras clases deben implementar. Sin embargo, cada una tiene características específicas y se utiliza en contextos diferentes.

1. **Clases Abstractas**

Una **clase abstracta** es una clase que no puede ser instanciada directamente, y se utiliza para proporcionar una estructura base para otras clases. Puede contener:

- Métodos abstractos (sin implementación), que deben ser implementados por las clases derivadas.
- Métodos concretos (con implementación), que pueden ser utilizados o sobrescritos por las clases derivadas.

```
abstract class Figura
{
    public abstract double CalcularArea(); // Método abstracto
    public void Descripcion()
    {
        Console.WriteLine("Soy una figura.");
    }
}
```

En este ejemplo, Figura es una clase abstracta con un método abstracto CalcularArea() que debe ser implementado por cualquier clase que herede de ella.

Ejercicio: Clase Abstracta

Crea una clase abstracta llamada Figura, que contenga un método abstracto CalcularArea(). Luego, crea dos clases derivadas: Cuadrado y Círculo. Cada una de estas clases debe sobrescribir el método CalcularArea() y calcular el área correspondiente.

- Clase “Figura”

```
using System;

abstract class Figura
{
    // Método abstracto
    public abstract double CalcularArea();

    // Método concreto
    public void Descripcion()
    {
        Console.WriteLine("Soy una figura.");
    }
}
```

- Clase “Cuadrado”

```
class Cuadrado : Figura
{
    public double Lado { get; set; }

    public Cuadrado(double lado)
    {
        Lado = lado;
    }

    // Sobrescribe el método abstracto
    public override double CalcularArea()
    {
        return Lado * Lado;
    }
}
```

- Clase “Circulo”

```
class Circulo : Figura
{
    public double Radio { get; set; }

    public Circulo(double radio)
    {
        Radio = radio;
    }

    // Sobrescribe el método abstracto
    public override double CalcularArea()
    {
        return Math.PI * Radio * Radio;
    }
}
```

- Programa Principal

```
class Programa
{
    static void Main()
    {
        Cuadrado cuadrado = new Cuadrado(4);
        Circulo circulo = new Circulo(5);

        Console.WriteLine($"Área del cuadrado: {cuadrado.CalcularArea()}");
        Console.WriteLine($"Área del círculo: {circulo.CalcularArea()}");
    }
}
```

Explicación:

- Figura es una clase abstracta que tiene un método abstracto CalcularArea() y un método concreto Descripcion().

- Cuadrado y Circulo heredan de Figura y sobrescriben el método CalcularArea() para calcular el área del cuadrado y del círculo, respectivamente.

Salida esperada:

```
Área del cuadrado: 16
Área del círculo: 78.53981633974483
```

2. Interfaces

Una **interfaz** es un contrato que define métodos, propiedades o eventos que una clase debe implementar. Las interfaces no pueden contener implementación de métodos, solo las firmas.

```
interface IVolador
{
    void Volar(); // Método sin implementación
}
```

En este ejemplo, IVolador es una interfaz que define un método Volar(). Cualquier clase que implemente esta interfaz deberá proporcionar una implementación para ese método.

Ejercicio: Interfaz

Crea una interfaz llamada IVehiculo que contenga dos métodos: Arrancar() y Detener(). Luego, crea dos clases Coche y Motocicleta que implementen esta interfaz.

- Clase “IVehiculo”

```
using System;

interface IVehiculo
{
    void Arrancar();
    void Detener();
}
```

- Clase “Coche”

```
class Coche : IVehiculo
{
    public void Arrancar()
    {
        Console.WriteLine("El coche ha arrancado.");
    }

    public void Detener()
    {
        Console.WriteLine("El coche se ha detenido.");
    }
}
```

- Clase “Motocicleta”

```
class Motocicleta : IVehiculo
{
    public void Arrancar()
    {
        Console.WriteLine("La motocicleta ha arrancado.");
    }

    public void Detener()
    {
        Console.WriteLine("La motocicleta se ha detenido.");
    }
}
```

- Programa Principal

```
class Programa
{
    static void Main()
    {
        IVehiculo coche = new Coche();
        IVehiculo moto = new Motocicleta();

        coche.Arrancar();
        coche.Detener();

        moto.Arrancar();
        moto.Detener();
    }
}
```

Explicación:

- IVehiculo es una interfaz que define los métodos Arrancar() y Detener().
- Coche y Motocicleta implementan la interfaz, proporcionando su propia versión de los métodos definidos en la interfaz.

Salida esperada:

```
El coche ha arrancado.
El coche se ha detenido.
La motocicleta ha arrancado.
La motocicleta se ha detenido.
```

Diferencias entre Clases Abstractas e Interfaces

Característica	Clase Abstracta	Interfaz
Instanciación	No se puede instanciar	No se puede instanciar
Métodos	Puede tener métodos concretos y abstractos	Solo tiene métodos abstractos
Herencia múltiple	No permite herencia múltiple	Permite implementar múltiples interfaces
Modificadores de acceso	Puede tener modificadores de acceso (public, protected, private)	Todos los métodos son public
Constructores	Puede tener constructores	No puede tener constructores

Ejercicio Clases Abstractas e Interfaces

Crea una clase abstracta InstrumentoMusical con un método abstracto Tocar(). Luego, crea una interfaz IAfinable con un método Afinar(). Implementa ambas en las clases Guitarra y Piano.

Conclusión

- Las **clases abstractas** permiten definir un comportamiento base que puede ser parcialmente implementado, y las clases derivadas lo pueden extender o sobrescribir.
- Las **interfaces** definen contratos que las clases deben seguir, pero no proporcionan ninguna implementación.
- Mientras que las **clases abstractas** permiten heredar de una sola clase base, las **interfaces** permiten implementar múltiples contratos, facilitando la creación de comportamientos más flexibles y modulares.

INSTITUTO

KHIPU