UTN F r a
Avellaneda

Programación y Laboratorio II

Clase 13
Interfaces

Temario

Interfaces

- ¿Qué es una interfaz?
- Usos y generalidades.

Implementación de interfaces

- Definición e Implementación.
- Interfaces Explícitas.

01.

Interfaces

¿Qué es una Interfaz?

Una interfaz es un medio común que nos permite agrupar **funcionalidades** que una clase luego debe implementar.

Se puede decir que las interfaces establecen un **contrato** en el cual las clases que implementan la interfaz están **obligadas** a implementar sus funcionalidades.

Interfaces

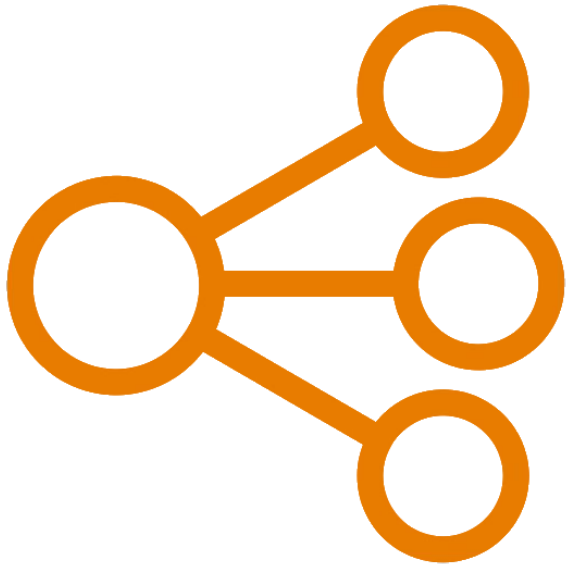
Como las interfaces nos permiten tener la funcionalidad separada de la implementación, se pueden tener distintas implementaciones en diferentes clases sin una relación fuerte, pero con la misma funcionalidad.

Enviar Mensaje

Interfaces

Usos y generalidades

- Nos permiten definir métodos, propiedades, indexadores y eventos.
- Una interfaz **no** permite definir atributos ni constructores.
- Una clase puede implementar **varias** interfaces.
- Todas las funcionalidades definidas en una interfaz son públicas, sin posibilidad de modificación,



02.

Implementación de interfaces

Definición e Implementación

```
public interface IMensaje
{
  string EnviarMensaje();
}
```

- Para definir una interfaz: se declara su modificador de visibilidad seguido de la palabra reservada **interface** y el identificador.
  Por convención, este siempre comienza con la letra **I** mayúscula.

- Los miembros definidos no llevan modificador de visibilidad.
  Por defecto van a ser siempre públicos.

Para implementar una interfaz: luego del identificador de la clase se emplea el operador **":"** seguido del nombre de la interfaz.

```
public class Email:IMensaje
{
    public string EnviarMensaje()
    {
      return "Ya falta mucho para implementar? :)";
    }
}
```

Si queremos implementar una interfaz en una clase derivada, la clase base siempre va a ir primero y luego irá la interfaz o interfaces separadas por una coma **","**.

```
public class Carta :Papel,IMensaje
{
    public string EnviarMensaje()
    {
        return "Esta es una declaracion de guerra";
    }
}
```

Interfaces Explícitas

Usos y generalidades

- Si una clase implementa más de una interfaz que contiene miembros con la misma firma, la llamada a ese miembro implementado, ocultará la implementación de los otros miembros de las otras interfaces.

- Usando la implementación explícita de una interfaz podemos indicarle al compilador el miembro de que interfaz vamos a implementar y vamos a poder proveer diferentes funcionalidades a los miembros de estas distintas interfaces.

- Del lado de la interfaz no vamos a ver reflejado ningún cambio ya que los cambios son en las implementaciones de estos métodos.

- Otra razón para utilizar interfaces explícitas es ocultar la implementación de los elementos de la interfaz para que no sean fácilmente accesibles.

Interfaces Explícitas

Para lograr esto, colocamos delante del nombre del miembro el nombre de la interfaz que estamos implementando (no se debe indicar el modificador de visibilidad).

```
public class Cuervo :IMensaje, IEncriptado
 {

     public string EnviarMensaje()
     {
         return "Llego el invierno";
     }

     string IEncriptado.EnviarMensaje()
     {
         return "Jon Snow es el verdadero rey";
     }
 }
```

← Forma Implícita

← Forma Explícita

```
static void Main(string[] args)
{
  Cuervo cuervo = new Cuervo("Bloodraven");

  //Casteamos al tipo de interfaz
  ((IEncriptado)cuervo).EnviarMensaje()

}
```

Para invocar al miembro implementado explícitamente debemos castear nuestro objeto al tipo de la interfaz.