

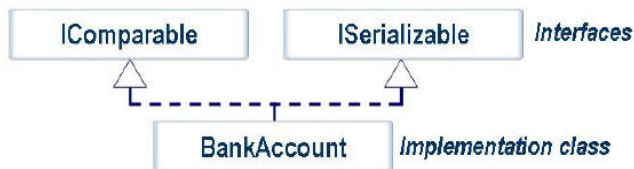


Programación II

Interfaces

- La interfaz de una clase define el comportamiento de dicha clase.
- Una interfaz define un conjunto de propiedades y métodos a ser implementados por otras clases.
- A diferencia de las clases, las interfaces no proporcionan implementación.
- Una interfaz representa un contrato.
- Las interfaces son implementadas por clases y se definen en entidades separadas.

Interfaces



```
public class BankAccount : IComparable, ISerializable
{
    public void GetObjectData(SerializationInfo info, StreamingContext context) {
        throw new NotImplementedException();
    }

    public int CompareTo(object obj) {
        throw new NotImplementedException();
    }
}
```

Beneficios:

- Desacoplamiento
- Gestión de dependencias
- Herencia múltiple de interfaces

Desacoplamiento

- Las interfaces permiten desacoplar la aplicación de las implementaciones concretas de las clases.
- En la aplicación podemos referirnos a un tipo de interfaz en lugar de una clase exacta y de esta forma trabajar con cualquier clase que implemente la interfaz.

Desacoplamiento

Problema:

```
static void Problema()  
{  
    Impresora1 impresora = new Impresora1();  
    impresora.Imprimir("Hola Mundo");  
}
```

¿Qué ocurre si se desea cambiar de impresora?

Desacoplamiento

Solución:

```
static void Solucion()  
{  
    IImpresora impresora = ObtenerImpresoraPreDeterminada();  
    impresora.Imprimir("Hola Mundo");  
}  
  
private static IImpresora ObtenerImpresoraPreDeterminada()  
{  
    return new Impresora1();  
}
```

Gestión de dependencias

- Las interfaces permiten reducir las dependencias de la aplicación.
- Las interfaces son generalmente más estables que las clases que los implementan.

Gestión de dependencias

Problema:

```
static void Imprimir(Impresora impresora)
{
    impresora.Imprimir("Hola Mundo");
}
```

Solución:

```
static void Imprimir(IImpresora impresora)
{
    impresora.Imprimir("Hola Mundo");
}
```

Herencia múltiple de interfaces

- Una clase puede heredar de varias interfaces.
- La clase debe implementar todos los miembros definidos en cada una de las interfaces que hereda.

```
public class Impresora1 : IImpresora, IDisposable
{
    public void Imprimir(string documento) {
        Console.WriteLine("{0}. [Documento impreso con Printer1]", documento);
    }

    public void Dispose() {
        Console.WriteLine("Elimino la impresora");
    }
}
```

Ejercicios

1. Revisar la definición de la clase encargada de generar los niveles del juego.
2. Construir la interfaz IGeneradorNiveles e implementarla en dicha clase.
3. Crear otros "generadores de niveles" implementando la nueva interfaz.