Instituto Tecnológico del sur de Nayarit



Maestra:

Ing. Cinthia Anahí Mata Bravo

Alumna:

Carelia Abigail Patiño Hernández

Semestre: 2

Carrera: tecnologías de la información y comunicaciones

Índice

Portada………………………………………………………pág. 1

Investigación…………………………………………………pág. 3-5

Resumen……………………………………………………. pág. 6

**Interfaces en C#**

Una técnica imprescindible para lograr un adecuado nivel de abstracción es la utilización de interfaces.

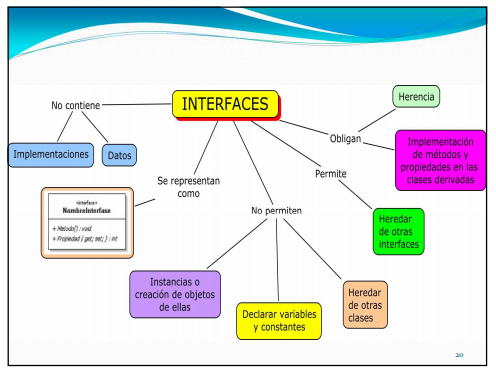
Hablando en llano, una interfaz no es más que una estructura de datos que muestra únicamente las firmas de los métodos de una clase. A partir de ahí, una clase que herede de la interfaz estará obligada a “rellenar” la implementación de dichos métodos. De forma simple, una interfaz sería algo así:

Interfaz

Motor  
Arrancar();  
Detener();

Esta interfaz únicamente dice que acciones se van a realizar, pero no CÓMO se realizarán. Para saber cómo realizarlas, crearemos clases que implementen esta interfaz, obligando a “rellenar” el contenido de los métodos que declaramos previamente.

La interfaz se puede comparar con una clase abstracta en la cual la cual puede tener varias implementaciones, pero en las interfaces no se pueden hacer implementaciones, en la comparación del lado de la herencia en una clase abstracta se puede hacer una herencia simple, en una interfaz puede hacer herencias multiples,en los modificadores de acceso en una clase abstracta puede tener modificadores de acceso y en una interfaz automáticamente sus miembros son públicos, en los miembros validos en las clases abstractas son: campos,propiedades,constructores,destructores,métodos,eventos,índices etc, en las interfaces:propiedades,métodos,eventos,índices etc.



Una interfaz define un contrato. Cualquier clase o estructura que implemente ese contrato debe proporcionar una implementación de los miembros definidos en la interfaz. Una interfaz puede definir una implementación predeterminada para los miembros. También puede definir miembros estáticos para proporcionar una implementación única para una funcionalidad común.

Una interfaz puede ser miembro de un espacio de nombres o una clase. Una declaración de interfaz puede contener declaraciones (firmas sin ninguna implementación) de los siguientes miembros:

Métodos

Propiedades

Indexadores

Eventos

Estas declaraciones de miembros anteriores generalmente no contienen un cuerpo. Los miembros con cuerpos permiten que la interfaz proporcione una implementación "predeterminada" para clases y estructuras que no proporcionan una implementación primordial. Puede contener

Constantes

Operadores

Constructor estático.

Tipos anidados

Campos estáticos, métodos, propiedades, indexadores y eventos.

Declaraciones de miembros que usan la sintaxis de implementación de interfaz explícita.

Modificadores de acceso explícito (el acceso predeterminado es público).

Las interfaces pueden no contener estado de instancia. Si bien los campos estáticos ahora están permitidos, los campos de instancia no están permitidos en las interfaces. Las propiedades automáticas de las instancias no son compatibles con las interfaces, ya que declararían implícitamente un campo oculto. Esta regla tiene un efecto sutil en las declaraciones de propiedad. En una declaración de interfaz, el siguiente código no declara una propiedad implementada automáticamente como lo hace en una clase o estructura. En cambio, declara una propiedad que no tiene una implementación predeterminada, pero debe implementarse en cualquier tipo que implemente la interfaz:

Una interfaz puede heredar de una o más interfaces base. Cuando una interfaz anula un método implementado en una interfaz base, debe usar la sintaxis explícita de implementación de la interfaz. Cuando una lista de tipos base contiene una clase base e interfaces, la clase base debe ser la primera en la lista.

Una clase que implementa una interfaz puede implementar explícitamente miembros de esa interfaz. No se puede acceder a un miembro implementado explícitamente a través de una instancia de clase, sino solo a través de una instancia de la interfaz. Además, solo se puede acceder a los miembros predeterminados de la interfaz a través de una instancia de la interfaz.

**Resumen:**

En esta investigación realizada entendí que el concepto de interfaz es una estructura de datos que muestra los métodos de clase en que se está utilizando, en donde las interfaces no permiten algunas cosas como lo es las instancias o creación de objetos de ellas, en lo que también cabe destacar que no se permite heredar de otras clases, pero si permite heredar de otras interfaces, en las interfaces es necesario usar o implementar métodos y propiedades en las clases que se usen, en donde también puede contener modificadores de acceso, otra de las cosas que no se puede hacer es que cuando se hace una interfaz no se puede acceder a sus miembro implementado en una instancia de clase, si no que a su instancia de interfaz.

En conclusión, para mí la definición queda como una interfaz define un contrato que la clase cumplirá. La interfaz define cuáles son las operaciones que puede realizar la clase.