

CLASES ABSTRACTAS VS. INTERFAZ

Clases abstractas

Clases cuya descripción es incompleta. Una clase abstracta declara métodos, pero no tiene que implementarlos.

No proporcionan la implementación de todos sus métodos

Los métodos no implementados se declaran como abstract

Una clase con un método abstracto debe declararse como clase abstracta

Pero una clase puede declararse como abstracta aunque no tenga ningún método abstracto

Las subclases de una clase abstracta deben:

Sobreescribir todos los métodos abstractos de la superclase, o bien ser declarada como clases abstractas

Una clase abstracta no pueden instanciarse

No se pueden crear objetos de una clase abstracta

Una clase abstracta puede incluir variables y métodos no abstractos.

No se pueden definir constructores abstractos ó métodos estáticos abstractos.

Interfaces

Sólo declaran comportamiento

Se utiliza la palabra clave interface

Por defecto todos sus métodos son públicos y abstractos

No implementan el comportamiento

Por defecto todos sus atributos son públicos, constantes y de clase

Por legibilidad normalmente los declaramos static y final

Permite simular algunos aspectos de la herencia múltiple

Define un tipo de datos

Posibilita el enlace dinámico . Otras clases pueden implementar un interfaz

Cualquier clase que implemente un interfaz debe definir todos los métodos de dicha interfaz

Debe proporcionar la implementación de dichos métodos

Si la clase no proporciona la implementación para todos los métodos del interfaz debe ser declarada como abstracta

Las interfaces sirven para:

Declarar métodos que serán implementados por una o más clases.

Determinar la interface de programación de un objeto, sin mostrar el cuerpo de la clase.

Capturar similitudes entre clases no relacionadas, sin forzar una relación entre ellas.

Describir objetos "tipo función", que podrán ser utilizados como argumentos al invocar métodos sobre objetos.

Clase abstracta vs Interfaz

Adicionalmente, me gustaría detenerme un poco en la diferencia entre una clase abstracta y una interfaz.

Veo con mucha frecuencia que muchas veces no queda del todo claro.

Y es que mucha gente no conoce bien la diferencia entre una clase abstracta y una interfaz, o en su caso, las confunde... así que permitidme que me detenga aquí un momento para enumerar las coincidencias y diferencias entre ambas.

Interfaz

Como hemos comentado, una interfaz se encarga de definir comportamientos, contratos. En cada interfaz, podemos declarar métodos, funciones, eventos, delegados o propiedades.

Todos los miembros de una interfaz son públicos y abstractos.

En otro orden de cosas, todo lo que declaremos en una interfaz, deberá ser implementado en las clases que implementen la interfaz.

Una clase por su parte, puede implementar más de una interfaz.

Finalmente, una interfaz no posee estado (*data members*) o implementación alguna (funciones y métodos).

Clase abstracta

Una clase abstracta puede contener funciones y métodos que serán utilizados por las clases que implementen la clase abstracta.

En cada clase abstracta, podemos declarar métodos, funciones, eventos, delegados, propiedades y variables.

Una clase abstracta puede incluir funcionalidad, al contrario que una interfaz.

Una clase sólo puede implementar una única clase abstracta, ya que no existe soporte para herencia múltiple.

Independientemente de todo lo comentado hasta ahora, una clase que herede de una clase abstracta no tiene porqué implementar ningún método abstracto de la misma ya que como clase derivada, es en sí misma una clase abstracta.

Finalmente, una clase abstracta puede contener estados (*data members*) e implementaciones (funciones y métodos).

En términos de velocidad o rendimiento, una clase abstracta es más rápida que una interfaz, ya que la clase que implementa la interfaz, debe preparar todas las definiciones de la misma.

➤ VER: <http://www.slideshare.net/networksens/09-clases-abstractas-e-interfaces>