Clase abstracta

Como indica su nombre, se trata de una clase, hasta ahí todo claro, pero que posee, he aquí su nombre y principal característica, al menos un método abstracto, por lo que ya tiene que ser declarada como abstracta. Una clase abstracta no puede ser implementada, sino que sólo puede heredar a otra clase abstracta o a una clase que defina el método o los métodos abstractos de la superclase abstracta.

Herencia

Interfaz

Una interfaz tiene la característica principal de que todos sus métodos son abstracto, es decir, todos los métodos son declarados pero ninguno tiene implementación. Al no ser una clase como tal, el resto de clases pueden implementar tantas interfaces como se estime oportuno.

Esta diferencia es sobre la herencia - cualquier clase puede implementar muchas interfaces, pero se puede extender una sola clase y puede tener un solo padre.

Clases múltiples que se extienden es una característica/funcionalidad que está presente solo en algunos lenguajes orientados a objetos. ¿Por qué? Porque trae más problemas que valor.

Cuando una clase tiene muchos padres y hay una situación que tenemos exactamente la declaración del mismo método en más de una, tendríamos que explícitamente "decir" cuál es el que nos interesa.

Visibilidad del Método

Todos los métodos de la interfaz son públicos, pero no hay tal regla para los declarados en las clases abstractas. Aquellos metodos pueden tener ninguna visibilidad excepto privado. ¿Por qué no privado? Debido a que un método abstracto debe implementarse en una subclase y si es privado no es accesible desde la subclase. Como podemos ver, estas dos ideas son mutuamente excluyentes.

Declaración vs. Definición

Una interfaz solo puede tener las declaraciones de los metodos y una clase abstracta puede o no contener la definición de los métodos.

Es así porque la interfaz se centra en proporcionar una funcionalidad específica y la clase abstracta puede enfocarse también en la similitud en las implementaciones de las subclases, no sólo por su funcionalidad.

Constantes

Tanto en las interfaces y en las clases abstractas no hay ningún problema con la definición de valores constantes. Es porque esos valores no dependen de un objeto en particular, sino que son los mismos para todos.

Atributos

Las clases abstractas pueden contener atributos, mientras que las interfaces no pueden. La razón es la misma que con declaraciones y definiciones.