Clases Abstractas en Java

¿Qué es una clase abstracta?

Habrá ocasiones en las cuales necesitemos crear una clase padre donde únicamente coloquemos la estructura de una abstracción, una estructura muy general, dejando que sean las clases hijas quienes definan los detalles.

En estos casos haremos uso de las clases abstractas.

Una clase abstracta es prácticamente idéntica a una clase convencional o concreta; las clases abstractas pueden poseer atributos, métodos, constructores, etc.

Una clase de la que no se tiene intención de crear objetos, sino que únicamente sirve para unificar datos u operaciones de subclases, puede declararse de forma especial en Java como clase abstracta.

Métodos abstractos

Un método abstracto tiene estas peculiaridades:

a) No tiene cuerpo (llaves): sólo consta de signatura con paréntesis.

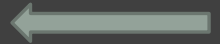
b) Su signatura termina con un punto y coma.

c) Sólo pueden existir dentro de una clase abstracta. De esta forma se evita que haya métodos que no se puedan ejecutar dentro de clases concretas. Visto de otra manera, si una clase incluye un método abstracto, forzosamente la clase será una clase abstracta.

d) Los métodos abstractos forzosamente habrán de estar sobre-escritos en las subclases. Si una subclase no implementa un método abstracto de la superclase significa que tiene un método no ejecutable (Java no lo permite), lo que la fuerza a ser una subclase abstracta. Para que la subclase sea concreta habrá de implementar métodos sobre-escritos para todos los métodos abstractos de sus superclases.

e) La utilidad de un método abstracto es definir qué se debe hacer pero no el cómo se debe hacer

```
public abstract double calcularArea();
```

 El método no está implementado!!!

Preguntas frecuentes I 

¿Es necesario que una clase que tiene uno o más métodos abstractos se defina como abstracta?

Sí, si declaramos un método abstracto el compilador nos obliga a declarar la clase como abstracta porque si no lo hiciéramos así tendríamos un método no ejecutable dentro de una clase concreta, y eso no es admitido por Java.

¿Una clase se puede declarar como abstracta y no contener métodos abstractos?

Sí, una clase puede ser declarada como abstracta y no contener métodos abstractos. En algunos casos la clase abstracta simplemente sirve para efectuar operaciones comunes a subclases sin necesidad de métodos abstractos.

Preguntas frecuentes II 

¿Una clase que hereda de una clase abstracta puede ser no abstracta?

Sí, de hecho esta es una de las razones de ser de las clases abstractas. Una clase abstracta no puede ser instanciada, pero pueden crearse subclases concretas sobre la base de una clase abstracta, y crear instancias de estas subclases. Para ello hay que heredar de la clase abstracta y anular los métodos abstractos, es decir, implementarlos.

¿Puede una clase abstracta tener constructor?¿Con qué fin?

Sí, Las clases abstractas siempre tienen un constructor, pero SOLAMENTE para ser usados desde los constructores de las clases hijas, no puedes usarlos directamente porque por definición no se puede instanciar una clase abstracta.

Si no especificamos uno entonces se le crea uno por defecto y sin argumentos, como ocurre con cualquier otra clase.

Preguntas frecuentes III 

¿Cuándo usar una clase abstracta y cuándo usar una interface?

Las clases abstractas pueden proporcionar abstracción parcial o total. Las interfaces, por otro lado, siempre proporcionan una abstracción completa. Se puede crear una clase padre abstracta para algunas clases que tienen algunas funcionalidades comunes. También se prefieren las clases abstractas si desea más libertad de acción.

Se prefieren las interfaces cuando queremos definir una estructura básica. El programador puede entonces construir cualquier cosa con esta estructura. Las interfaces también admiten múltiples herencias. Entonces, una sola clase puede implementar múltiples interfaces.

En general, es una cuestión de elección y la tarea que se debe realizar. Tanto la clase abstracta como la interfaz son adecuadas para diferentes propósitos y deben usarse en consecuencia.

¿Puede una clase abstracta implementar una interface?

Sí, ¿porqué no?. Una clase abstracta no deja de ser una clase normal de Java aunque con determinadas características.