**CLASES ABSTRACTAS**

Clases “molde” de las cuales se puede heredar, pero nunca se pueden crear objetos de dicha clase, aunque sí se pueden declarar objetos de ese tipo. Una clase es abstracta si:

* Contiene un método abstracto (sin implementación, solo su declaración), el cual puede ser heredado.
* Si va precedida de la palabra abstract. En tal caso, podría incluso no tener métodos abstractos.

*Ejemplo 01 Eclipse*

**INTERFACES**

Una interfaz es

***Diferencias entre interfaces y clases abstractas***

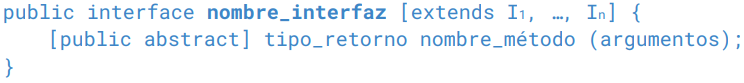
Interfaces:

* Todos los métodos de una interfaz se declaran implícitamente abstractos y públicos.
* No puede implementar ningún método.
* No tiene atributos salvo que sean estáticos o constantes, e implícitamente siempre son públicos.
* No puede pertenecer a una jerarquía de clases.
* Solo puede extender a otras interfaces.

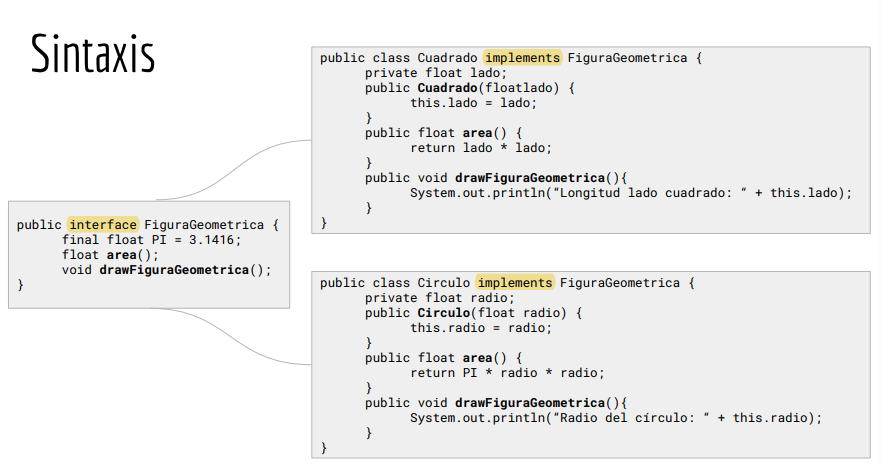
Clases abstractas:

* La clase abstracta sí puede implementar métodos.
* Puede implementar varias interfaces, pero solo puede extender a una clase.
* Puede pertenecer a una jerarquía de clases. Por lo tanto, clases no relacionadas pueden implementar la misma interfaz.
* Una clase puede extender otras clases e implementar distintas interfaces.

Su sintaxis es la siguiente:



Deben declararse en un archivo con el mismo nombre y extensión .java.



***Implementación***

Una clase puede implementar una o varias interfaces. Implementar una interfaz significa implementar métodos de la interfaz. Si la clase no implementa algún método de la interfaz, entonces la clase es necesariamente abstracta. Además, puede tener métodos propios. Si no implementa algún método, entonces es abstracta.

Para poder utilizar una constante declarada en una interfaz, o un atributo estático, es necesario anteponer el nombre de la interfaz a la constante.

La jerarquía entre interfaces permite la herencia simple y múltiple.

* Una clase puede implementar varias interfaces.
* Una interfaz puede “extender” a otras interfaces, pero no puede implementarlas.
* Una interfaz no puede tener métodos estáticos (un método estático no puede ser abstracto).

Una clase puede, simultáneamente, heredar de otra clase e implementar una o varias interfaces. Una interfaz es un tipo, por lo tanto, el tipo se puede utilizar en cualquier lugar donde se pueda utilizar el identificador de un tipo o de una clase.