

Existen varias diferencias entre una clase abstracta y una interfaz:

- Una clase abstracta **puede heredar o extender cualquier clase** (independientemente de que esta sea abstracta o no), mientras que una interfaz solamente puede **extender o implementar otras interfaces**.
- Una clase abstracta **puede heredar de una sola clase** (abstracta o no) mientras que una interfaz puede **extender varias interfaces** de una misma vez.
- Una clase abstracta **puede tener métodos que sean abstractos o que no lo sean**, mientras que las interfaces **sólo y exclusivamente pueden definir métodos abstractos**.
- En java concretamente, en las clases abstractas la palabra **abstract es obligatoria para definir un método abstracto** (así como la clase). Cuando defines una interfaz, esta palabra es **opcional** ya que se infiere en el concepto de interfaz.
- En una clase abstracta, los **métodos abstractos pueden ser public o protected**. En una interfaz **solamente** puede haber **métodos públicos**.
- En una clase abstracta pueden existir **variables static, final o static final** con cualquier modificador de acceso (public, private, protected o default).
- En una interfaz **sólo** puedes tener **constantes** (public static final).

Si se necesita programar una clase de la que vayan a heredar otras, pero que esas otras clases que heredan, compartan alguna funcionalidad (ejemplo: clase abstracta -> Persona. Clases que heredan > Alumno, Profesor... Todas pueden tener un atributo nombre y métodos get y set de este atributo que hagan lo mismo), en ese caso es recomendable utilizar una clase abstracta. En caso contrario, interfaz.