

Clases Abstractas vs Interfaces en Java

Similitudes

Se utilizan para la abstracción.

No se pueden instanciar, es decir, no podemos crear un objeto para ellas.

Las subclases deben sobrescribir los métodos abstractos definidos en ellas

Diferencias

| Clases Abstractas | Interfaces |
|--|---|
| La palabra clave abstract en Java se utiliza para crear o declarar una clase abstracta. | En Java, la palabra clave interface se utiliza para crear o declarar una nueva interface. |
| Una clase puede heredar las propiedades y los métodos de una clase abstracta utilizando la palabra clave extends. | Para implementar una interface en Java, usaremos la palabra clave implements. |
| Una clase abstracta puede tener métodos abstractos o no abstractos definidos en ella. | Una interfaz solo puede tener métodos abstractos. Después de Java 8, también podemos tener métodos static implementados. |
| Una clase abstracta puede contener variables finales o no finales (atributos de clase). También puede contener atributos estáticos o no estáticos. | Una interface solo puede contener miembros estáticos y finales. |
| Una clase abstracta puede implementar una interface e implementar los métodos de la interface. | Una interface no puede extender una clase ni implementar métodos. |
| Una clase abstracta puede ampliar otras clases y también puede implementar interfaces. | Las interfaces no pueden extender otras clases. Pero no hay ninguna restricción para que extiendan interfaces. |
| Java no admite herencias múltiples a través de clases. Las clases abstractas, al igual que cualquier otra clase, no admiten herencias múltiples. | El soporte para múltiples herencias en Java se proporciona a través de interfaces. Esto se debe a que las interfaces proporcionan una abstracción completa. |
| Los miembros o atributos de la clase abstracta pueden ser privados, protegidos o públicos. | Los atributos o miembros de una interface son siempre públicos. |