

1- Para qué se utilizan los genéricos

Los genéricos son un conjunto de características del lenguaje Java que permiten al programador usar tipos y funciones genéricos y así garantizar la seguridad de los tipos.

Por ejemplo, en un ArrayList (genérico) podemos guardar cualquier tipo de objeto en su interior, especificando en cada caso, el tipo.

2- Podemos hacer o mismo con la clase Object ¿? de ser así que ventajas tienen los genéricos

Sí se puede hacer lo mismo con la clase Object

Los genéricos garantizan la seguridad de tipo ya que la verificación de estos se hace en tiempo de compilación y no de ejecución.

También tiene como ventaja la reutilización de código ya que no es necesario escribir un código separado para cada tipo de datos, se puede escribir una sola clase, método, etc y usarlo para todos los tipos de datos.

3- ¿Qué son los comodines .?

Los comodines para tipos genéricos son la solución específica para un problema muy concreto: leer y escribir en colecciones genéricas.

Se indica con un signo de interrogación el cual se utiliza para referirse a un tipo desconocido. Los comodines se utilizan principalmente con genéricos como tipo de parámetro.

4- ¿Sólo podemos implementar métodos genéricos .?

No, también puedes crear clases e interfaces genéricas

5- ¿Cómo indicamos que un método es genérico.?

Con "<T>", por ejemplo:

```
public class Bolsa < T > implements Iterable < T > {  
    private ArrayList < T > lista = new ArrayList < T > ();
```

6- Cando debemos utilizar unha clase que implemente as seguintes interfaces .?

- Set: Es una colección que no puede contener elementos duplicados. Esta interfaz contiene, únicamente, los métodos heredados de Collection añadiendo la restricción de que los elementos duplicados están prohibidos.
- Queue: Queue es una interfaz que extiende a Collection y proporciona operaciones para trabajar con una cola, en donde, el primero en entrar es el primero en salir.
- Map: Este tipo de colección está formado por una clave, la cual no se repite, y de un valor, que sí puede repetirse.
- Pila: Al contrario de Queue, el primero de entrar es el ultimo en salir(LIFO).