
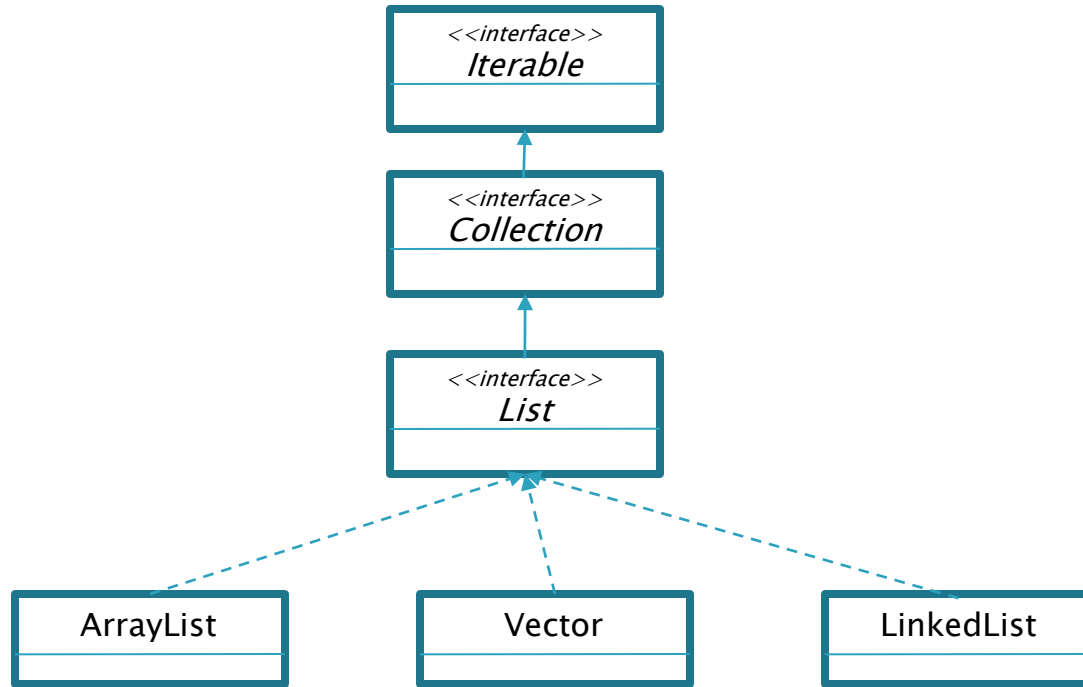


Colecciones

Listas

- Cada elemento tiene una posición asociada a partir del orden de llegada, siendo 0 la posición del primero
 - Las listas implementan la interfaz List, que a su vez implementa Collection.
 - Son colecciones de tipo genérico (preparadas para admitir cualquier objeto Java)
 - La principal clase de colección es ArrayList.
- 

Classes e interfaces de Listas



Creación de listas

➤ Como instancias de ArrayList:

```
List<Integer> enteros=new ArrayList<>();
```

➤ A partir del método asList de Arrays:

```
List<Integer> enteros=Arrays.asList(6,2,4,10,21);
```

Tamaño FIJO, no admite añadir ni eliminar

➤ Mediante método de factoría de List:

```
List<Integer> enteros=List.of(40,29,11,28);
```

INMUTABLES, no admiten la eliminación, modificación e inserción de elementos

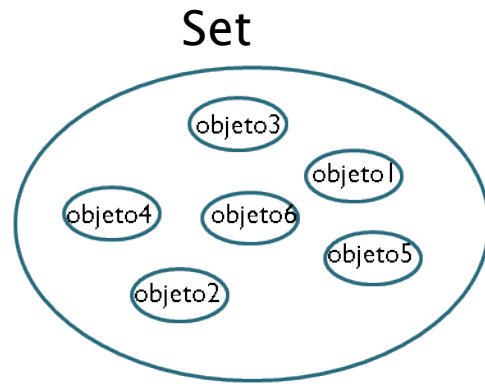
➤ Mediante el método copyOf de List:

```
List<Integer> copia=List.copyOf(enteros);
```

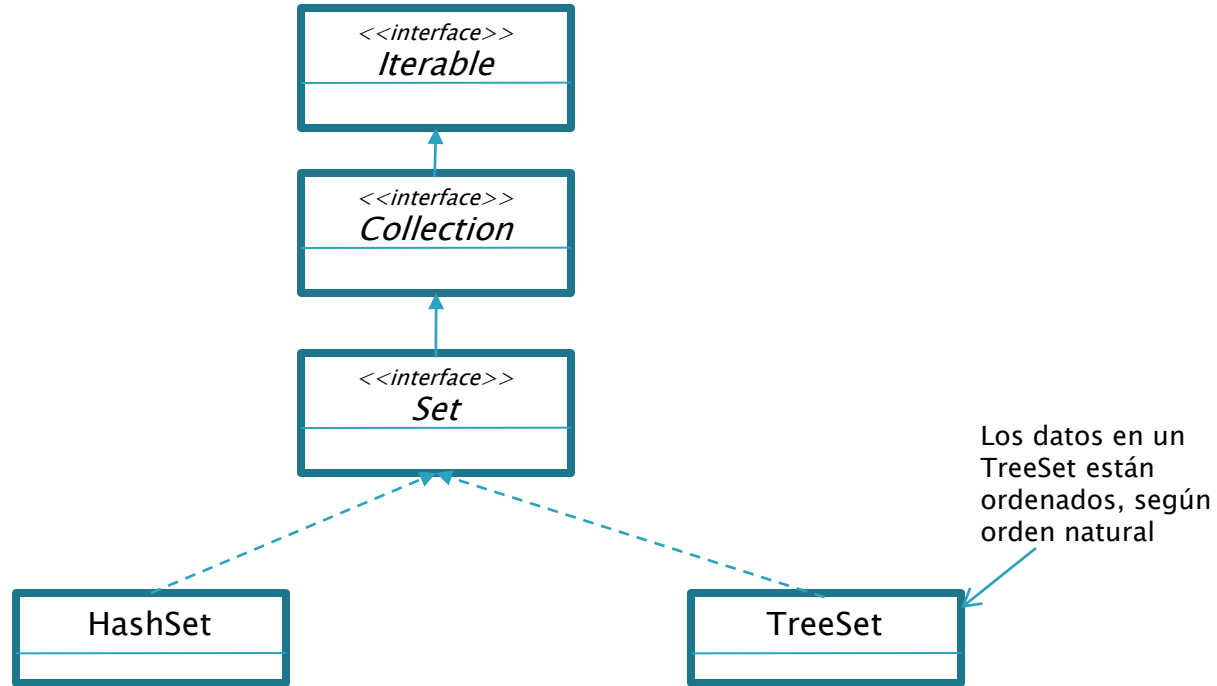
No admite valores null

Conjuntos

- Los elementos no tienen posición ni clave asociada, si bien cada elemento es único, no se pueden repetir
- Emplea internamente los métodos equals y hashCode para determinar la igualdad de objetos
- Los conjuntos implementan la interfaz Set, que es de tipo genérico
- La principal clase de conjuntos es HashSet



Clases e interfaces de conjuntos



Creación de conjuntos

➤ Como instancias de HashSet:

```
Set<String> nombres=new HashSet<>();
```

➤ Mediante método de factoría de Set:

```
Set<String> nombres=Set.of("María", "Luis", "Alberto");
```

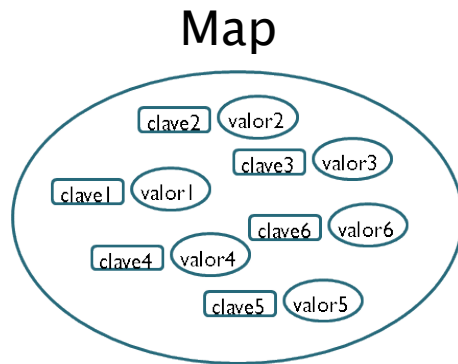
➤ Mediante el método copyOf de Set:

```
Set<String> nuevos=Set.copyOf(nombres);
```

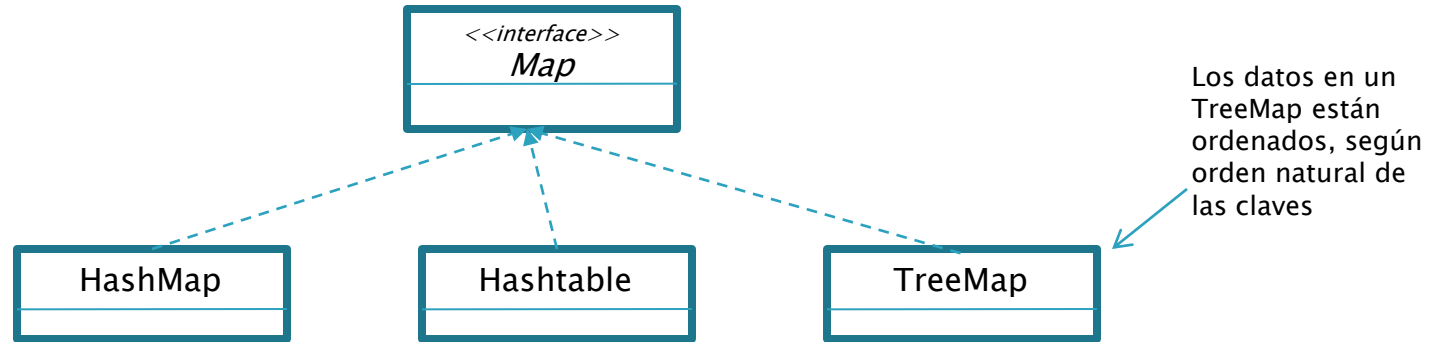
INMUTABLES, no admiten la eliminación, modificación e inserción de elementos.
No admite valores null

Tablas

- Cada elemento tiene asociada una clave única
- No hay un orden o posición
- Las tablas implementan la interfaz Map.
- Tanto el tipo de la clave como del valor son genéricos
- La principal clase de colección es HashMap.



Clases e interfaces de tablas



Creación de tablas

➤ Como instancias de HashMap:

```
Map<Integer, String> contactos=new HashMap<>();
```

➤ Mediante método de factoría de Map:

```
Map<Integer, String> contactos=Map.ofEntries(  
    Map.entry(123, "Luis"),  
    Map.entry(300, "Ana"),  
    Map.entry(500, "Maria"));
```

INMUTABLES, no admiten la eliminación, modificación e inserción de elementos. No admite ni claves ni valores null

➤ Mediante el método copyOf de Map:

```
Map<Integer, String> nuevos=Map.copyOf(contactos);
```

Sobre herencia

➤ Si B hereda A, no implica que una colección de B sea subtipo de una colección de A:

```
A dato=new B();
```

```
List<A> datos=new ArrayList<B>(); //error de compilación
```