

COLECCIONES Y TIPOS ABSTRACTOS DE DATOS

Interface Collection	
Hereda de Iterable	
int size()	nº de elementos contenidos
boolean isEmpty()	indica si está vacía
void clear()	vacía la colección
boolean contains(Object o)	retorna <i>true</i> si el elemento ya existe
boolean containsAll(Collection <?> c)	retorna <i>true</i> si la colección ya existe
boolean add(Object o)	retorna <i>true</i> si el elemento se añade
boolean addAll(Collection <?> c)	retorna <i>true</i> si se añade algún elemento de la colección
boolean remove(Object o)	retorna <i>true</i> si el elemento se elimina
boolean removeAll(Collection <?> c)	retorna <i>true</i> si se elimina algún elemento de la colección
boolean retainAll(Collection <?> c)	Mantiene todos los elementos de c retorna <i>true</i> si se elimina algún elemento de la colección
Iterator<Object> iterator()	Retorna un iterador sobre la colección
Object[] toArray()	Retorna un array con los elementos de la colección

Interface Set Hereda de Collection	
HashSet	<ul style="list-style-type: none">• Es la collection Set más rápida• No hay orden
LinkedHashSet	<ul style="list-style-type: none">• Orden de entrada
TreeSet	<ul style="list-style-type: none">• Orden ascendente• No puede contener null

Interface List Hereda de Collection Hay duplicación	
ArrayList	<ul style="list-style-type: none">• Array dinamico• Permite valores null
LinkedList	<ul style="list-style-type: none">• Permite valores null

Collection ArrayList Hereda de List	
boolean add(int posicion, E elemento)	Añade elementos
boolean addAll(int posicion, Collection <?> c)	Añade colecciones
E get(int posicion)	Devuelve el elemento en la posición determinada
int indexOf(Object o)	Devuelve la primera posición de un elemento
int lastIndexOf(Object o)	Devuelve la última posición de un elemento

E remove(int posicion)	Elimina elementos de una posición determinada
E set(int posicion, E elemento)	Reemplaza elementos en una posición
sort(Comparator<?> c)	Ordena los elementos del array

Interface Map conjuntos clave-valor no admite repeticiones	
HashMap	<ul style="list-style-type: none">• Tabla Hash• Admite <i>nulls</i>
LinkedHashMap	<ul style="list-style-type: none">• Orden de inserción
TreeMap	<ul style="list-style-type: none">• Más lento• Ordenación por la interfaz <i>Comparable</i>

Interface Map conjuntos clave-valor no admite repeticiones	
void clear()	Vacía el mapa
boolean containsKey(Object clave)	Devuelve <i>true</i> si la clave existe
boolean containsValue(Object valor)	Devuelve <i>true</i> si el valor existe
Object get(Object clave)	Devuelve el valor asociado a la clave
boolean isEmpty()	Devuelve <i>true</i> si el mapa está vacío

Object put(Object clave, Object valor)	Inserta una entrada
Object remove(Object clave)	Elimina la entrada
V replace(K clave, V valor)	Sustituye el valor asociado a la clave pasada por argumento
int size()	Devuelve el número de entradas