Bloque 3

Colecciones y Mapas

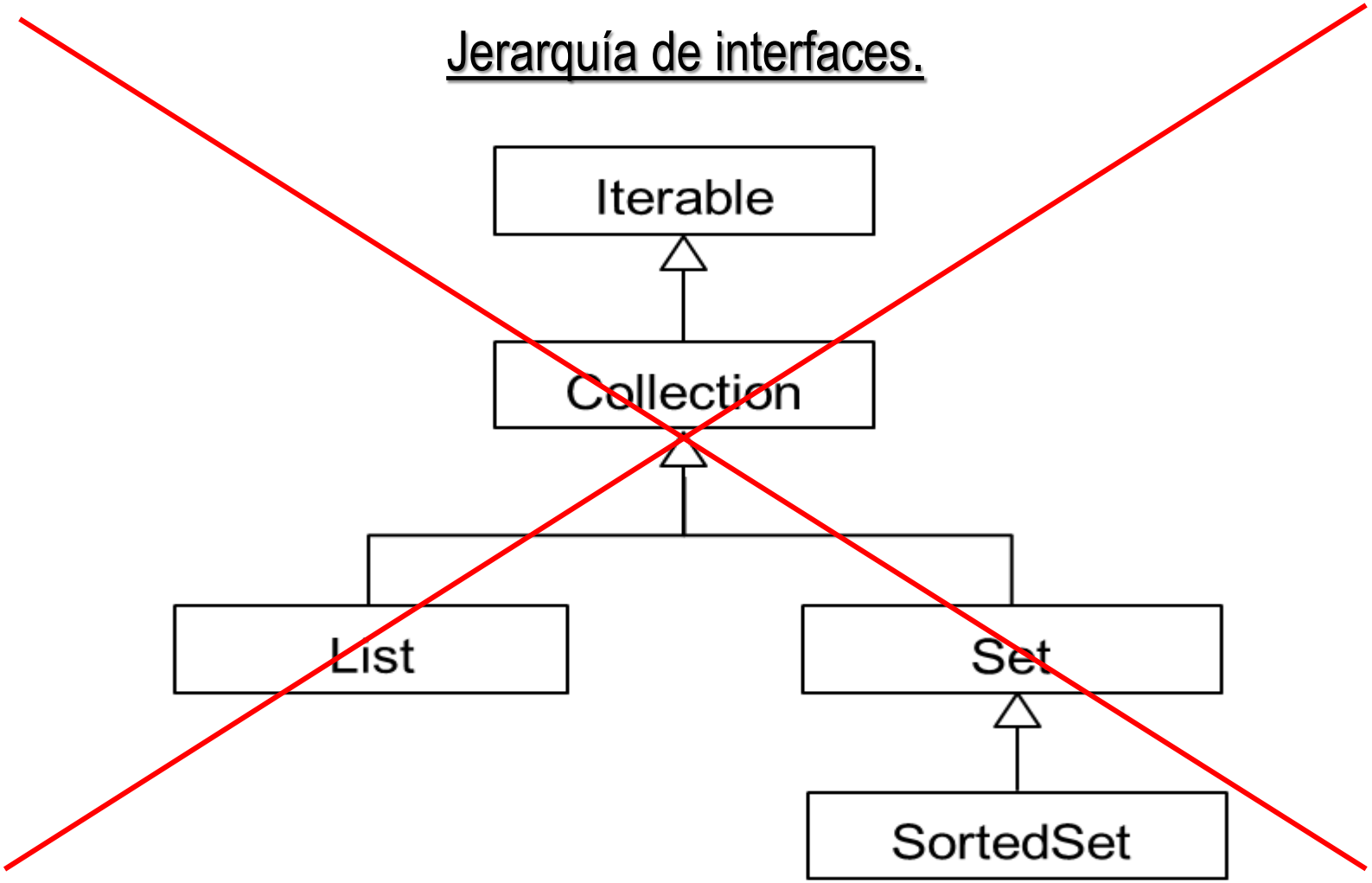
Collections

Fundamentos de Programación
Departamento de Lenguajes y Sistemas Informáticos

2023/2024

2023/24

Introducción a Collections



- Colecciones: Listas, Conjuntos y Conjuntos Ordenados

Introducción a Collections

Clase de Utilidad Collections (*introducción*).

- En primer lugar, no confundir, esta clase ***Collections*** (que termina en "s") con la interfaz ***Collection*** (sin "s").
- Mientras que ***Collection*** es sólo una interfaz de las que heredan las interfaces ***List*** y ***Set*** que ya hemos visto, ***Collections*** es una ***clase de utilidad*** con *métodos para complementar* el manejo de colecciones.
- Es una clase del paquete *java.útil*
- Al ser una clase de utilidad todos sus métodos son ***estáticos*** por lo que, para usarlos necesitan que sean invocados con el nombre de la clase: ***Collections***.*nombreMetodo*(…colección/es..)

Introducción a Collections

Clase de Utilidad Collections (*métodos*).

Algunos de los métodos más usados son los que se relacionan a continuación, aunque en:

"*https://docs.oracle.com/javase/7/docs/api/java/util/Collections.html*" están todos los métodos de dicha clase.

Métodos más comunes:

- static <E> *boolean addAll* (Collection<? super E> **c**, E... **elementos**). Añade a la colección "**c**" cada uno de los **elementos** que se indiquen a partir del segundo parámetro separados por coma ",". Devuelve *true* si c es modificada. Los elementos a añadir deben ser del tipo de objetos que almacena la colección "**c**" <E> o subtipos de ellos <? Super E>.

Introducción a Collections

Clase de Utilidad Collections (*métodos*).

- static <E extends Object & Comparable<? super E>> *E max* (Collection<? extends E> **c**). Devuelve el elemento máximo de la colección "**c**" según el orden natural de sus elementos.
- static <E> *E max*(Collection<? extends E> **c**, Comparator<? super E> **comparador**). Devuelve el elemento máximo de la colección c según el orden inducido por el objeto **comparador** que se pasa con segundo parámetro.

Introducción a Collections

Clase de Utilidad Collections (*métodos*).

- static <E extends Object & Comparable<? super E>> *E min* (Collection<? extends E> **c**). Devuelve el elemento mínimo de la colección "**c**" según el orden natural de sus elementos.
- static <E> *E min*(Collection<? extends E> **c**, Comparator<? super E> **comparador**). Devuelve el elemento mínimo de la colección c según el orden inducido por el objeto **comparador** que se pasa con segundo parámetro.

Introducción a Collections

Clase de Utilidad Collections (*métodos*).

- static *void reverse* (List<?> l).Invierte la posición de los elementos de la lista "**l**".
- static *void shuffle* (List<?> l). Mezcla aleatoriamente (baraja) los elementos de la lista "**l**".
- static <T extends Comparable<? super T>> *void sort*(List<T> **l**). Ordena la lista "**l**" según el orden natural del tipo.
- static <T> *void sort*(List<T> **l**, Comparator<? super T> **comparador**) Ordena la lista "**l**" según el orden inducido por el objeto **comparador.**

Observar que estos cuatro métodos no devuelven nada (son de tipo *void*) por lo que modifican las listas que reciben como parámetro perdiendo la ordenación original.

Si se quiere conservar la original hay que guardar una copia.

2023/24

Ejercicio Aeropuerto

- Realice los ejercicios del ***EnunciadoAeropuerto05***

  Apartados 14, 15 y 16