

# ORDENACIÓN

## COLECCIONES

# ORDENACIÓN

- Java proporciona algunas colecciones ordenadas
  - Inserción: ArrayList, LinkedList, LinkedHashXXX
  - Según valor: TreeXXX
- También podemos ordenar listas y arrays.

# ORDENACIÓN DE ARRAYS

- Métodos *Arrays.sort* y *Arrays.parallelSort*
- Implementaciones para los tipos primitivos y Object
- Implementación genérica si se proporciona un orden a través de una instancia de *Comparator*.
- Todos los métodos permiten una ordenación parcial del array a través de dos índices.

# ORDENACIÓN DE LISTAS

- A través del método *Collections.sort()*.
- Recibe como un argumento objeto de tipo *List*
  - *ArrayList*, *LinkedList*, ...
  - Los objetos de la lista tienen que implementar *Comparable*
- Modifica la colección original.
- Versión del método aceptando un *Comparator*.

# ORDENACIÓN DE OTRAS COLECCIONES

- Si usamos una colección que no sea List podemos utilizar las versiones ordenadas del resto de colecciones
  - *Set* → *TreeSet*
  - *Map* → *TreeMap*
- Ambas tienen un constructor que acepta un *Comparator* para ordenar usando otro diferente al *orden natural*.