

Estructuras de Datos

Tarea Iteradores y Comparadores

Profesora: Adriana Collaguazo

Los siguientes métodos deben ser implementados utilizando Iteradores y Comparadores. Cuando no se especifique, usted es libre de decidir los parámetros y tipos de dato de retorno de los métodos indicados. Cuando lo considere pertinente, incluya comentarios en sus respuestas para indicar cualquier consideración especial en cuanto a sus soluciones. Así mismo, usted puede decir el iterable o comparador que crea pertinente.

1. **public int countNegatives(Iterable<Integer> numbers):** Cuenta cuántos números en un Iterable son negativos.
2. **public int countStringsLongerThan(Iterable<String> strings, int length):** Cuenta cuántas cadenas en un Iterable tienen una longitud mayor a length.
3. **public void sortDescending(List<Integer> list, Comparator<Integer> comparator):** Ordena la lista en orden descendente usando el Comparator.
4. **public void sortByLastLetter(List<String> strings, Comparator<String> comparator):** Ordena la lista de strings de forma ascendente por la última letra de cada cadena usando el Comparator.
5. **public Iterable<Integer> skipEverySecond(Iterable<Integer> numbers):** Devuelve un iterable que omite cada segundo elemento [1, 2, 3, 4, 5] -> [1, 3, 5].
6. **public boolean areAllUnique(List<Integer> list, Comparator<Integer> comparator):** Verifica si todos los elementos de la lista son únicos según el criterio del Comparator.
7. **public Integer findMedian(List<Integer> list, Comparator<Integer> comparator):** Encuentra la mediana de una lista utilizando el Comparator para ordenar y determinar el valor central (Cantidad de elementos -> Impar: la mediana valor del medio exacto, Par: la mediana es el promedio de los dos valores centrales).
8. **public Iterable<Integer> takeWhileLessThan(Iterable<Integer> numbers, int threshold):** Devuelve un iterable con los elementos de numbers hasta que aparezca uno mayor o igual a threshold.
9. **public Iterable<String> filterStringsStartingWith(Iterable<String> strings, char prefix):** Filtra strings que empiecen con el carácter prefix y devuélvelos en un nuevo Iterable.
10. **public int binarySearch(List<Product> list, Product target, Comparator<Product> comparator):** En base a una lista de Productos (pueden definir la clase como prefieran), el método retorna el índice en donde se encuentra el producto (target), si el producto no está en la lista retornará -1.