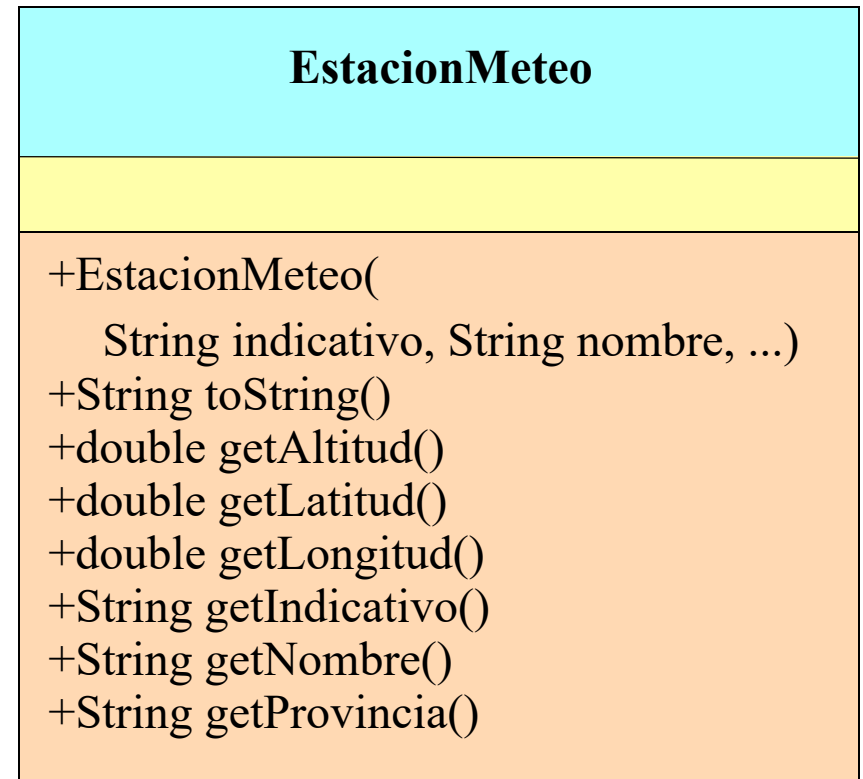


# Práctica 11: Lista de Estaciones Meteorológicas

**Objetivos.** Practicar con las tablas y la escritura de datos en ficheros

Se dispone de una clase que contiene datos básicos de una estación meteorológica, de acuerdo con el diagrama de clase que se muestra en la figura



# Clase ListaEstaciones

Se desea escribir la clase `ListaEstaciones`, que contiene un array con una lista de objetos de la clase `EstacionMeteo`, de acuerdo al diagrama que se muestra

- el constructor se proporciona ya implementado
  - rellena la lista a partir de datos predefinidos, obtenidos de la AEMET

Es preciso escribir los demás métodos

`numEstacionesSuperioresA():`

retorna el número de estaciones cuya altitud supera el parámetro `altura`

ListaEstaciones
-EstacionMeteo[] lista
+ListaEstaciones() +listado(String nombreFichero) +int numEstacionesSuperioresA (double altura) +ArrayList<EstacionMeteo> obtenerEstacionesEnZona (double latInferior, double lonIzqda, double latSuperior, double lonDcha) +String nombreEst(String indicativo)

# Clase ListaEstaciones (cont.)

---

`listado()`: almacena en el fichero de texto cuyo nombre se indica en el parámetro una lista de todas las estaciones meteorológicas con su indicativo, nombre, provincia y altitud, una por línea. Los datos aparecen en columnas, con una descripción de cada columna en la primera línea. Ejemplo del formato deseado:

Indicativo	Nombre	Provincia	Altitud (m)
1387E	A CORUÑA AEROPUERTO	A CORUÑA	98
1387	A CORUÑA	A CORUÑA	58
1428	SANTIAGO DE COMPOSTELA AEROPUERTO	A CORUÑA	370
8175	ALBACETE BASE AÉREA	ALBACETE	702
8178D	ALBACETE	ALBACETE	674
8019	ALICANTE/ALACANT AEROPUERTO	ALICANTE	43
8025	ALICANTE/ALACANT	ALICANTE	81
63250	ALMERÍA AEROPUERTO	ALMERIA	21
6297	ALMERÍA	ALMERIA	7
90910	FORONDA-TXOKIZA	ARABA/ALAVA	513

...

Observar el alineamiento izquierdo del nombre y provincia

# Clase `ListaEstaciones` (cont.)

---

`obtenerEstacionesEnZona`: retorna un `ArrayList` con las estaciones comprendidas en la zona definida por el "rectángulo" comprendido entre las latitudes y longitudes indicadas

`nombreEst()`: retorna un `String` con el nombre de la estación cuyo indicativo se pasa como parámetro. Si no hay ninguna, retorna `null`

# Parte avanzada

---

**Objetivo:** aprender técnicas de ordenación de arrays

Añadir a la clase `ListaEstaciones` los siguientes métodos:

- `void ordenaPorAltitud()`: ordenar el array `lista` por las altitudes de las estaciones, de menor a mayor
- `void ordenaPorLatitud()`: ordenar el array `lista` por las latitudes de las estaciones, de Norte a Sur

**Nota:** buscar en internet formas de ordenar un array usando clases comparadoras (`Comparator`) y los métodos `sort` de `java.util.Arrays`. Por ejemplo:

- <http://darkbyteblog.wordpress.com/2011/03/31/java-arreglos-ordenamiento-de-objetos-con-comparable-y-comparator-clase-java-util-arrays/>

# Entregar

---

El proyecto Bluej en un archivo comprimido

Informe:

- Parte básica:
  - El código java de los métodos realizados en la clase `ListaEstaciones`
  - Los resultados de las pruebas indicadas en el informe
- Parte avanzada:
  - El código java de las nuevas clases y métodos
  - Los resultados de las pruebas indicadas en el informe