



### EJERCICIO 1 – TIPO BASE (1,5 puntos)

Se quiere realizar un estudio sobre las viviendas para ayudar a una agencia inmobiliaria en su labor diaria. Para ello, se dispone de un archivo `catalogo.csv`, situado en la carpeta `data` del proyecto que se le proporciona. El fichero tiene la siguiente estructura, donde la información de cada línea se corresponde con la superficie (en m<sup>2</sup>), el precio (en euros), el número de habitaciones, el tipo de vivienda, la dirección, y el distrito postal:

```
72.26,114454.50,2,CHALET,Calle del Cerrillo,41001
179.07,175847.51,5,DUPLEX,Calle de la Constitucion,41005
188.50,104049.55,5,PISO,Plaza de la Constitucion,41006
...
```

Teniendo en cuenta que en el proyecto tiene definido el tipo enumerado **TipoVivienda**, que puede tomar los valores PISO, ADOSADO, PAREADO, CHALET, DUPLEX, se pide implementar la clase **Vivienda** teniendo en cuenta la siguiente descripción.

#### Propiedades:

- superficie, de tipo Double. Consultable. Superficie en m<sup>2</sup>.
- precio, de tipo Double. Consultable. Precio en euros.
- precio por metro cuadrado, de tipo Double. Consultable. Precio por metro cuadrado en euros.
- número de habitaciones, de tipo entero. Consultable. Número de habitaciones de la vivienda.
- tipo, de tipo TipoVivienda. Consultable. Tipo de la vivienda.
- direccion, de tipo String. Consultable. Dirección de la vivienda.
- distrito, de tipo String. Consultable. Distrito postal en el que está situado la vivienda.
- capital, de tipo Boolean. Consultable. Es true si el tercer dígito del distrito postal es cero, y false en caso contrario.

Constructores: un solo constructor que tiene como parámetros las propiedades básicas del tipo, y en el mismo orden en el que se han definido en la descripción de las propiedades anteriores.

Criterio de igualdad: dos viviendas son iguales si tienen la misma dirección y el mismo distrito postal.

Criterio de ordenación: por dirección, y a igualdad de dirección, por distrito.

Representación como cadena: el valor de todas las propiedades, separados por coma.

#### Restricciones:

- La superficie y el precio deben ser mayores que cero.
- El número de habitaciones debe ser mayor o igual que cero.
- El distrito postal debe estar compuesto por 5 dígitos.

### EJERCICIO 2 – FACTORÍA (1,5 puntos)

Teniendo en cuenta que en el proyecto se le proporciona el tipo contenedor `Catalogo` parcialmente implementado, implemente el tipo **FactoriaCatalogo** para poder cargar la información del fichero de datos proporcionado. Añádale los métodos que crea convenientes para distribuir correctamente las responsabilidades, e implemente los test necesarios para comprobar que la lectura de fichero funciona correctamente.

### EJERCICIO 3 – TIPO CONTENEDOR (7 puntos)

Implemente el tipo contenedor **Catalogo** y los test necesarios en la clase TestCatalogo para comprobar que las operaciones funcionan correctamente. La descripción del tipo es la siguiente:

#### Propiedades:

- viviendas, de tipo List<Vivienda>. Consultable. Lista con viviendas.

#### Constructores:

- Un constructor que tiene como parámetro un Stream<Vivienda> y crea un objeto de tipo Catalogo a partir del Stream.

Criterio de igualdad: dos objetos de tipo Catalogo son iguales si sus listas de viviendas son iguales.

Representación como cadena: la representación como cadena de la lista de viviendas.

#### Otras operaciones:

- a) **Map<String, Double> getPrecioMedioPorDistritoDeTipos(Set<TipoVivienda> conjTipos).** Recibe conjTipos, un conjunto de TipoVivienda, y devuelve un Map donde las claves son los distritos postales y los valores el precio medio de las viviendas de los tipos especificados en el conjTipos correspondiente a ese distrito. **(1 Punto)**
- b) **Set<String> getDistritosMasNViviendas (Integer n).** Dado un entero n, devuelve un conjunto con los distritos postales en los que hay más de n viviendas en el Catalogo. **(1,5 Puntos)**
- c) **List<TipoVivienda> getTiposViviendasMasCarasCalle (String calle, Integer n).** Recibe el nombre de una calle y un entero n, y devuelve una lista con los distintos tipos de las n viviendas con el precio por metro cuadrado más caro de esa calle. Esto es, se seleccionan las n viviendas con el precio por m<sup>2</sup> más caro y se devuelve una lista con los tipos distintos de esas viviendas. Se considera que una vivienda está en una calle si su dirección incluye el nombre de la calle. **(1,5 Puntos)**
- d) **Map<TipoVivienda, Vivienda> getViviendaMasBarataPorTipoEnCapital().** Devuelve un Map que relaciona el tipo de Vivienda con la vivienda más barata de ese tipo, solo para las viviendas situadas en la capital de Provincia. **(1,5 Puntos)**
- e) **Map<String, Double> getPorcViviendasPreciosMayorUmbralPorDistrito(Double umbralPrecio).** Recibe un umbral de precio de tipo Double y devuelve un Map tal que a cada distrito postal le haga corresponder el porcentaje de viviendas que en ese distrito supera en precio por m<sup>2</sup> el umbral dado. **(1,5 Puntos)**