

ESTRUCTURAS DE DATOS

Curso 2019/20

PRÁCTICA 2

Listas y Mapas

Instrucciones

- Se debe completar en dos sesiones.
- Práctica individual.
- Lee el enunciado completo antes de comenzar. Los comentarios incluidos en el código también proporcionan información útil y necesaria.
- Al finalizar la práctica debes entregar el código desarrollado.
- La práctica será APTA si se superan todos los test de validación proporcionados.

IMPLEMENTACIÓN DE POLINOMIOS MEDIANTE LISTAS

Un polinomio es una suma de monomios de distintos grados, por ejemplo, el polinomio

$$3x^3 - 2x^2 - 2$$

está compuesto por 4 monomios de grados 3, 2, 1, 0, respectivamente. Cada monomio tiene esta forma:

$$ax^b$$

dónde a es el coeficiente, x es la variable, y b (mayor o igual que 0) es el exponente de la variable o grado del monomio.

Hemos decidido implementar una clase que almacene un polinomio, para ello vamos a usar una lista en la que cada elemento almacene un monomio. Aplicaremos estas tres reglas:

1. No almacenaremos monomios cuyo coeficiente sea cero, así ahorramos espacio
2. No almacenamos monomios con el mismo grado.
3. Los monomios se almacenarán en la lista ordenados según el grado de menor a mayor.

De esta forma el polinomio anterior se representaría mediante esta lista.

$$[(-2, 0) (-2, 2) (3, 3)]$$

Cada par de valores entre paréntesis representa el coeficiente y el grado del monomio respectivamente. Proporcionamos tres ficheros con tres clases que implementan estos principios.

Monomio.java: Implementa la clase **Monomio** que representa cada uno de esos pares. La clase incluye constructores, y los métodos **equals** y **toString**.

Polinomio.java: Implementa la clase **Polinomio**, que incluye una lista de monomios como datos. Define diversos constructores, los métodos **equals** y **toString**. También define métodos para sumar polinomios, multiplicarlos por un monomio y proporcionar la lista de monomios.

Cero.java: Implementa una clase con métodos estáticos para determinar cuando un valor **double** es cero.

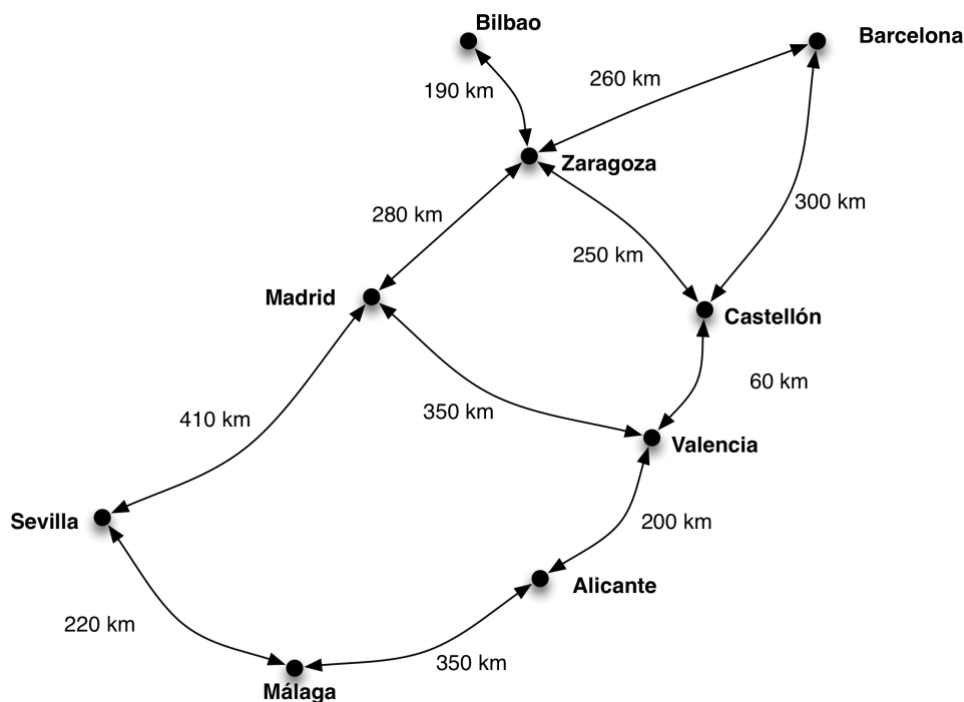
Ejercicio 1

Completa la implementación de la clase escribiendo los métodos:

```
public Polinomio(double v[])  
public void multiplicarMonomio(Monomio mono)  
public Polinomio sumar(Polinomio otro)
```

IMPLEMENTACIÓN DE POLINOMIOS MEDIANTE LISTAS

Deseamos implementar una clase que almacene una red de carreteras. Una red de carreteras se representa como un conjunto de ciudades y una serie de tramos que unen las ciudades por pares. Cada tramo une dos ciudades, es bidireccional y tiene asociada la distancia en kilómetros de dicho tramo



Para almacenar esta información decidimos implementar la clase **RedCarreteras** que internamente almacena los datos de la red en un diccionario con esta definición:

```
Map<String, Map<String, Integer>> red;
```

Se trata de un diccionario que tiene como clave un **String**, que almacena el nombre de una ciudad. Como valor asociado otro diccionario. Este segundo diccionario almacenará las ciudades a las que se puede llegar desde la primera, junto con las distancias de esos tramos. Así pues, y a partir del ejemplo, la entrada con clave "Castellón" almacenará un diccionario con los siguientes pares:

("Zaragoza", 250), ("Barcelona", 300) y ("Valencia", 60)

Ejercicio 3

Completa la implementación de la clase **RedCarreteras** implementando los métodos:

```
public RedCarreteras()  
public int nuevoTramo(String una, String otra, int distancia)  
public int compruebaCamino(List<String> camino)
```

Fíjate que la clase **RedCarreteras** ya implementa algunos métodos que te puede ser de utilidad.