

◁ Ejercicio 1 ▷ Rutas y tramos

[10 puntos]

Una **ruta** indica el camino que se ha de seguir para llegar a una ciudad *destino* partiendo de otra ciudad *origen* pasando por un número indeterminado de ciudades *intermedias*. Una ruta está formada por **tramos** y cada tramo está caracterizado por una ciudad *inicial*, una ciudad *final* y la distancia entre ambas.

Ejemplos de tramos serían: (Granada, Motril, 80.2), (Guadix, Granada, 62.5), etc. Un ejemplo de ruta sería: {(Guadix, Granada, 62.5), (Granada, Motril, 80.2)}

- No se permite un tramo en el que la ciudad inicial y la final sea la misma. Evidentemente, la distancia deberá ser estrictamente positiva.
- No se permiten cruces en rutas ni rutas con bucles: en definitiva, una ciudad aparece una sola vez en la ruta.

Implemente las clases Tramo y Ruta.

- En un Tramo nos interesa guardar los *nombres de las ciudades* (de tipo de dato *string*) y la *distancia* entre ambas (*double*, estrictamente positivo).
- En una Ruta queremos guardar una secuencia de datos Tramo. Si la secuencia tiene n tramos $(0, 1, 2, n-1)$, el tramo i ($i > 0$) debe verificar que su ciudad inicial sea la ciudad final del tramo $i-1$ y que su ciudad final no esté en ningún tramo anterior.

Implemente, al menos, lo siguiente:

1. Constructores de las clases Tramo y Ruta.
2. Métodos de consulta triviales (*get*) para las dos clases.
3. Métodos de cálculo para la clase Ruta:
 - (a) distancia total de la ruta.
 - (b) distancia entre dos ciudades, dados sus nombres.
4. Método para **añadir** un Tramo a una Ruta. Si el tramo no fuera correcto la ruta no se modifica.
5. Método para **sustituir** una serie de datos Tramo en una Ruta por un nuevo Tramo. Para que se realice la sustitución debe cumplirse que la distancia entre los tramos que se pretende sustituir sea mayor o igual que la del nuevo tramo. Si no es así, la Ruta no se modifica.

Por ejemplo, en esta ruta: ((C1,C2,10.5),(C2,C3,0.5),(C3,C4,2.6), (C4,C5,3.2), (C5,C6,1.5)) si se pretende sustituir en ella una serie de tramos por el tramo (C2,C4,2.5) sí será posible porque la distancia entre C2 y C3 más la distancia entre C3 y C4 ($0.5+2.6=3.1$) es mayor que la distancia entre C2 y C4 en el nuevo tramo (2.5). En este caso, la nueva ruta sería ((C1,C2,10.5),(C2,C4,2.5),(C4,C5,3.2), (C5,C6,1.5)).

Si sobre esa misma ruta se intentara sustituir por el tramo (C2,C4,3.5) no se haría nada, quedando la ruta sin cambios.

Puede definir los métodos auxiliares que estime oportuno.

Escriba una función *main* que lea los datos de una ruta y cree la ruta. Los datos para crear la ruta mostrada anteriormente se proporcionan como en el ejemplo siguiente:

C1 C2 10.5 C2 C3 0.5 C3 C4 2.6 C4 C5 3.2 C5 C6 1.5 #

Nota: Suponga que los datos serán correctos.

A continuación, el programa leerá los datos de un nuevo tramo, intentará sustituir ese tramo en la ruta y mostrará la ruta resultante junto a la distancia total.