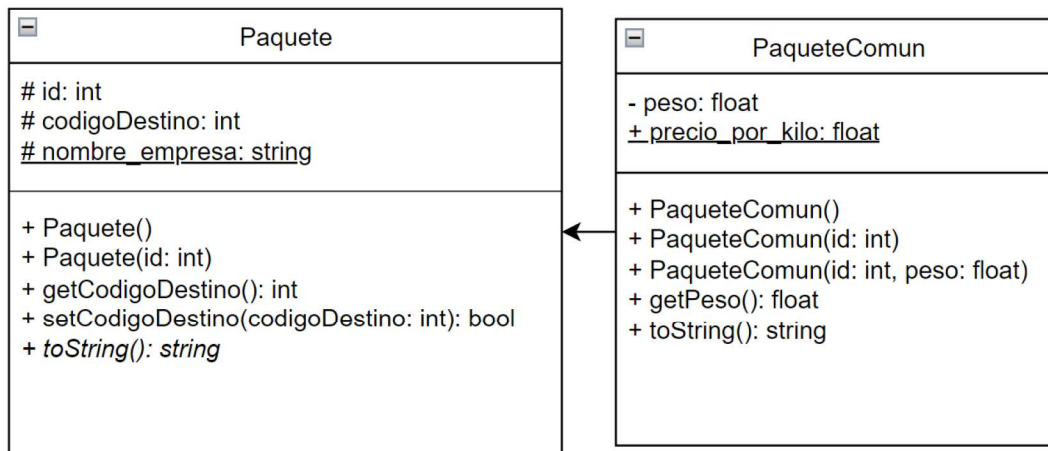


Ejercicios Tema 3. Construcción, Miembros estáticos, Composición, Herencia.

Problema 1.

Considere un servicio de entrega que gestiona paquetes. Para ello, se presenta el siguiente diagrama de clases:



Como se observa **PaqueteComun** hereda de **Paquete**, por lo que **PaqueteComun** *es-un* **Paquete**. Todo paquete contiene un número entero que representa su identificador (por ejemplo, un código de barras), un código de destino, es decir, el código postal a donde se tiene que dirigir el paquete y por último el nombre la empresa de entregas. El **PaqueteComun** se refiere a todo paquete que se transporte en volumen determinado pudieron tener diferentes pesos, por lo que todo **PaqueteComun** tiene un atributo `peso` y un `precio_por_kilo`.

Codifique ambas clases tales que cumplan con las siguientes características:

Paquete:

- Al llamar al constructor por defecto los atributos tipo enteros deben inicializarse a 0.
- Al utilizar el constructor que recibe un identificador, este debe almacenarse y el código de envío debe inicializarse a -1.
- El setter del código de destino debe validar que el nuevo destino tenga 5 caracteres, en este caso se almacena el nuevo valor y se devuelve true, en caso contrario no se almacena el nuevo valor y se devuelve false.
- El nombre de la empresa se definirá en el archivo principal en la memoria estática.

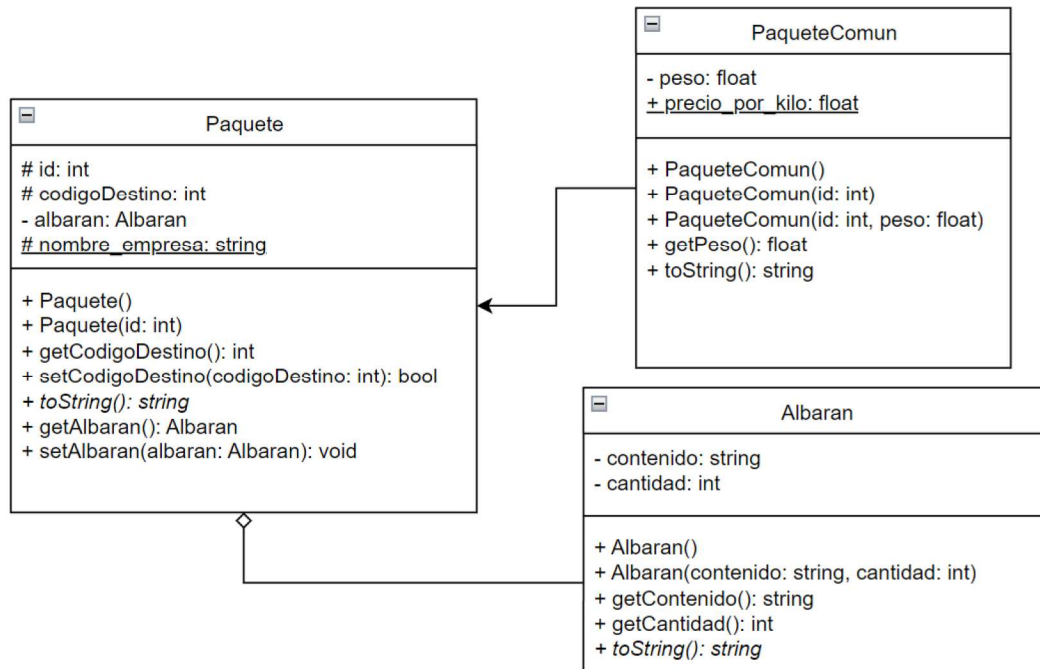
PaqueteComun:

- Todo Constructor de **PaqueteComun** deberá llamar al constructor equivalente de la clase superior, es decir que `PaqueteComun()` deberá llamar a `Paquete()` y `PaqueteComun(id: int, peso: float)` deberá llamar a `Paquete(id: int)`.
- El nombre de la empresa se definirá en el archivo principal en la memoria estática.

Problema 2.

A continuación, se requiere que todos los Paquetes integren un Albarán. Este documento deberá tener información de el contenido y la cantidad de los bienes. Por ejemplo, si un paquete consiste en una caja que contenga 6 Móviles, el albarán deberá almacenar Contenido: Móvil y Cantidad: 6.

De esta forma, el diagrama queda de la siguiente manera:



Albarán:

El albaran no tiene setters y solamente puede asignarse un contenido y una cantidad en el constructor que recibe esos datos.

Se debe modificar el `toString` de **Paquete** para agregar el `toString` de **albaran**.

Cree el programa principal que instancia un objeto **Paquete** y un objeto **PaqueteComun**, a cada objeto asigne un albarán diferente e imprima en pantalla los resultados.