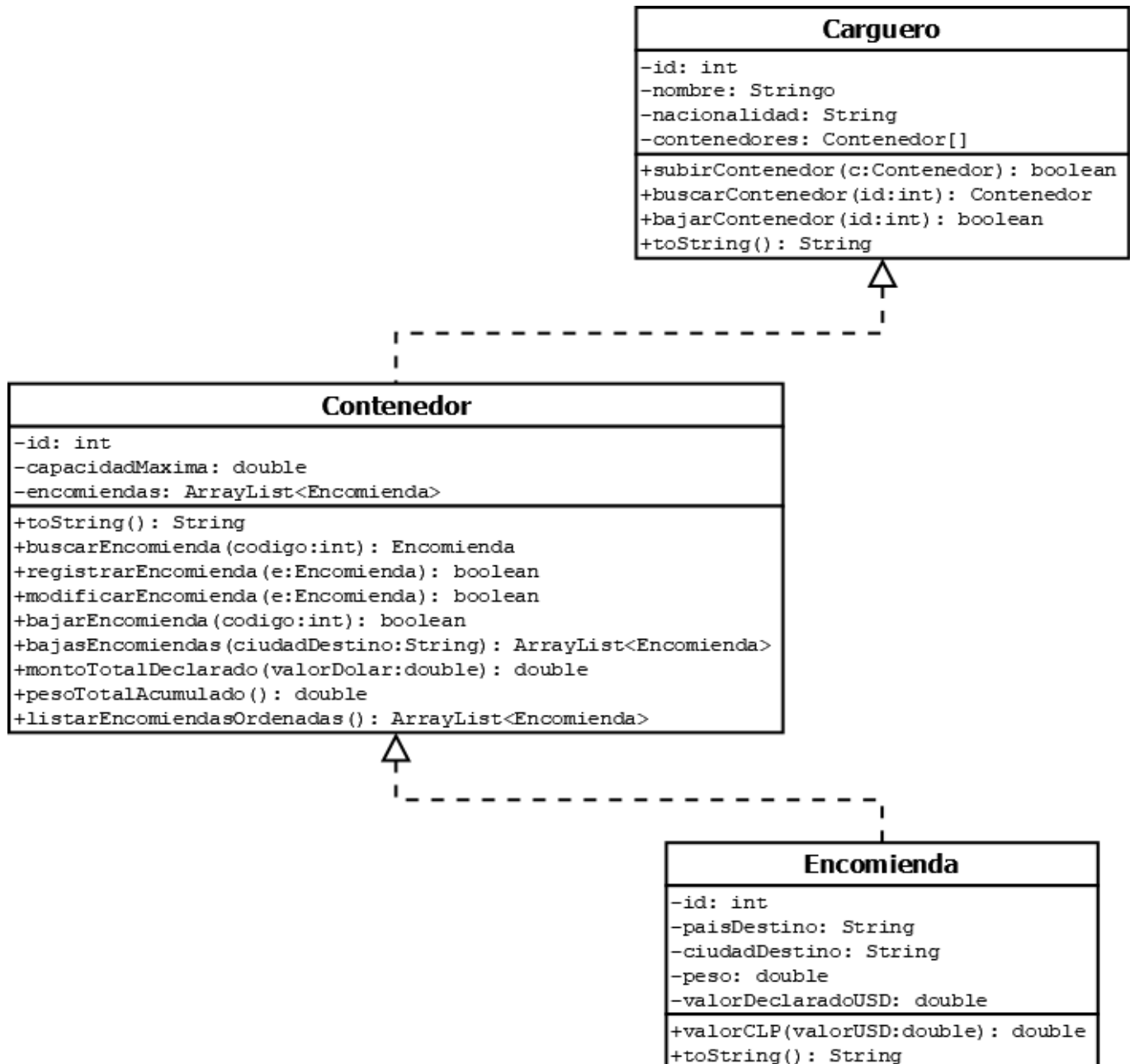


Ejercicio: POO – Colecciones ArrayList

Una empresa de transporte marítimo necesita administrar los contenedores y encomiendas que transportan sus cargueros, para ello le han solicitado a su equipo que desarrollen una solución para poder llevar a cabo esta magna operación

A continuación, se entrega el modelo de clases que representa en términos generales lo que se necesita.



CODIFICANDO LAS CLASES

Se solicita que para cada clase se definan constructor, accesadores y mutadores según sea necesario. Junto a ello se necesita que modifique el método toString de cada clase para que entregue la información cada clase de una manera clara y sencilla. En el caso de la clase Carguero, se debe entregar la cantidad de Contenedores que hay cargados. En el caso de la clase Contenedor, se debe entregar la cantidad de encomiendas.

Ejemplos:

Carguero: 12345 | Nombre: Nautilus | País: Chile | Nro. de contenedores: 2 / 10

Contenedor: 11111 | Capacidad: 1500 / 2000 | Cantidad Encomiendas: 2

Contenedor: 55555 | Capacidad: 0 /2000 | Cantidad Encomiendas: 0

Encomienda: 10001 | País Destino: Chile | Ciudad Destino: Santiago | Peso: 1000 | Valor Declarado: 15000 USD

Encomienda: 10002 | País Destino: Chile | Ciudad Destino: Valparaíso | Peso: 5000 | Valor Declarado: 5000 USD

Respecto a los métodos CUSTOM de cada clase, deberá analizar cuáles son los requerimientos necesarios para su desarrollo, es decir, que condiciones se deben cumplir para registrar, buscar, modificar o bajar (eliminar).

Clase Encomienda:

valorCLP: este método debe retornar el valor declarado en moneda nacional. El valor del dólar se recibe como parámetro y se deberá multiplicar el valorDeclaradoUSD por el valorUSD. El método deberá retornar el valor sin decimales.

Clase Contenedor:

buscarEncomienda: este método permite buscar una encomienda a través del código. En caso de encontrarla deberá retornar la Encomienda, de lo contrario deberá retornar NULL.

registrarEncomienda: este método permitirá almacenar una nueva encomienda en la Colección de encomiendas que posee esta clase. Se debe considerar que la encomienda no este registrada previamente. Además, se debe validar que el contenedor tenga la capacidad de poder almacenarla según el peso. Si el proceso de registro se completa satisfactoriamente deberá retornar TRUE, de lo contrario FALSE.

modificarEncomienda: este método permitirá modificar los datos de una encomienda que esté registrada en el contenedor. Para ello deberá buscar si es que está en caso de ser así deberá modificar la encomienda por la nueva encomienda que se desea modificar. El proceso retornará TRUE en caso de que se realice correctamente, FALSE en caso contrario.

bajarEncomienda: este método permitirá eliminar una encomienda del contenedor. El proceso retornará TRUE si la encomienda es eliminada correctamente. En caso contrario se retornará FALSE.

bajarEncomiendas: este método permitirá eliminar varias encomiendas del contenedor, para ello deberá buscar todas las encomiendas que son enviadas a la ciudad recibida como parámetro. El resultado de este método es un nuevo ArrayList de Encomiendas con todas las encomiendas que fueron eliminadas del ArrayList original. Si no hay encomiendas con dicho destino se retornará NULL.

montoTotalDeclarado: este método deberá retornar el total declarado por todas las encomiendas que contiene el contenedor. Este método recibe como parámetro el valor del dólar.

pesoTotalAcumulado: este método permite obtener el peso total de todas las encomiendas que tiene el contenedor. No puede superar jamás la capacidad máxima del contenedor.

listarEncomiendasOrdenadas: este método debe ordenar el ArrayList de Encomiendas por varios criterios. En primer lugar, por País de destino, luego por Ciudad de Destino, peso, valor declarado y finalmente por ID.

Clase Carguero:

subirContenedor: este método permitirá subir un contenedor al carguero. Para ello deberá contemplar que el contenedor no esté ya registrado o que no tenga espacios disponibles. El método retornará TRUE o FALSE según se logre subir o no.

buscarContenedor: este método permitirá buscar un contenedor a partir del código. El método debe retornar el contenedor o en su defecto NULL si no lo encuentra.

bajarContenedor: este método permitirá bajar o eliminar un contenedor del carguero. El método retornará TRUE o FALSE según el resultado de la operación.

Para probar el funcionamiento de su código se requiere que en la clase Main realice las siguientes operaciones

1. Crear un carguero, en el cual se defina que la cantidad de Contenedores sea 5.
2. Crear un contenedor
3. Crear 10 encomiendas
4. Luego realice el proceso de registrar las encomiendas en el contenedor y el contenedor en el carguero
5. Ejecute los métodos que permitan ver y verificar los datos ingresados, entre ellos el método

Recuerde documentar todo su código mediante el uso de JAVADOC y general la documentación del proyecto.

Utilice la nomenclatura del lenguaje para la definición de clases, atributos y encapsule todo su código.