**😃 Leé por lo menos dos veces el enunciado antes de resolver.**

# Enunciado

La compañía Terminales Argentinas S.A. nos pide desarrollar un sistema para sus terminales de autoservicio.

Estas terminales tienen la capacidad de permitir el depósito de billetes. Estos billetes son escaneados por la terminal reconociendo su número, año, letra de serie, valor (10, 50, 100) y estado (“BUENO”, “REGULAR”, ”MALO”) y, si cumple con una serie de requisitos, se almacenan uno arriba de otro. Todo billete debe poder ser verificable por la terminal. Los dólares, a diferencia de los pesos, tienen además una banda de seguridad 3D que indica también el valor del billete.

Los billetes pueden ser Nacionales (Pesos) o Dólares.

En el caso de los billetes nacionales, para cumplir con los requisitos de verificación, el año del billete debe ser posterior al 2010.

En el caso de los dólares, los requisitos de verificación implican que la letra de serie no puede ser ‘C’ y además el valor de la banda de seguridad 3D debe coincidir con el valor del billete.

En ambos casos, sean dólares o pesos, el número del billete debe ser superior a 2.000.000.

Basado en el enunciado descripto, realizá:

1. El diagrama de clases que lo modelice, con sus relaciones, atributos y métodos.
2. El método **depositar**, que recibe un Billete como parámetro y debe **devolver** si el mismo logró almacenarse en la terminal o no.
3. El método **cantBilletesPorEstadoYValor** que debe **devolver** (no mostrar por consola) la cantidad de billetes que hay almacenados en la terminal agrupados por estado y valor.
4. El método **listarBilletesOrdenadosPorValor** que debe **devolver** (no mostrar por consola) una lista de todos los billetes almacenados en la terminal ordenados por valor.

**IMPORTANTE:** En cada caso en el que se necesite recorrer la estructura donde se almacenan los billetes la estructura de datos donde se almacenan los billetes debe quedar en el orden original.

# Criterios

Para considerar aprobado el examen, el mismo debe demostrar la correcta aplicación de los siguientes conceptos de la programación orientada a objetos:

* Correcta definición de clases y asignación adecuada de sus responsabilidades.
* Encapsulamiento, ocultamiento de información y uso de getters y setters sólo cuando corresponda.
* Modularización reutilizable y mantenible con uso de métodos con correcta parametrización.
* Correcta aplicación de miembros de instancia y de clase.
* Correcta aplicación de herencia y polimorfismo.
* Correcta aplicación conceptual de las relaciones entre clases.
* Correcta aplicación de TADs vistas en clase

