

# Módulo Objetos

## Ejercicios Combinados

---

1- ARBA (Agencia de Recaudación de la Provincia de Buenos Aires), **dispone** de la información correspondiente al impuesto vehicular a pagar por los contribuyentes de la provincia de Buenos Aires. Cada **vehículo** posee su año de fabricación, importe de gestión básico a pagar por el impuesto y la información del propietario. Para cada uno de los **propietarios** se registra su CIT (Clave de Identificación tributaria), nombre y apellido.

Existen diferentes **tipos** de vehículos:

- **Automotor**, que también tiene asociado una patente, un importe adicional según el año de fabricación y una descripción.
- **Embarcación**, que tiene asociado un código del Registro Especial de Yates (REY), nombre, un tipo de embarcación (lancha, crucero, velero, etc.), eslora (longitud de proa a popa), un tonelaje (peso) y un valor declarado de la embarcación.

La Agencia necesita calcular los costos de los impuestos de la siguiente manera:

- De los **automotores**, el propietario paga mensualmente el importe básico más el adicional calculado según su año de fabricación.
- De las **embarcaciones**, el propietario paga un monto semestral que varía según el valor declarado de la embarcación: valuadas hasta \$6.000, pagan el 4% del su valor declarado; entre \$6.000 y \$180.000, pagan el 2% del valor declarado y, finalmente superiores a \$180.000 pagan el 5% de dicho valor declarado.

Se pide:

- a. Modelar e implementar la clase **Vehículo** con sus respectivos atributos, constructores y métodos para acceder y modificar atributos, y todo lo que considere necesario para representar los tipos de vehículo y para poder realizar los cálculos mencionados.
- b. Cree la clase **ProgramaArba**, su método main y los métodos que considere necesarios para la lectura y almacenamiento de 10000 vehículos. Luego, realice el cálculo del monto a pagar por el impuesto para cada uno de ellos.

2- Una editorial universitaria, necesita registrar la información correspondiente a los ejemplares que publicó y se publicarán. Cada **Ejemplar** posee un código identificador, cantidad de páginas, un resumen, año de publicación (si aún no fue publicado este valor es cero) y la información del responsable a cargo de la edición. Para el **responsable** se registra su DNI, nombre y apellido.

Existen diferentes tipos de ejemplares:

- **Libro**, que también tiene asociado un título, la cantidad de capítulos del mismo y si es o no una edición de bolsillo.
- **Revista**, que tiene asociado su nombre, un número de volumen y la cantidad de artículos en su interior.

Los ejemplares deben poder imprimirse en un listado que se genera periódicamente, por ello se pide que las mismas definan un mensaje que les permita **retornar un String** con sus datos:

- Para los **libros**, su código identificador, su título y el nombre del responsable.

- Para las **revistas**, su nombre y el número de volumen.

Por otro lado, cuando los ejemplares son publicados se agrega el año actual como año de publicación y, en particular para cada tipo de ejemplar:

- Para los **libros**, se concatena al final de su título si el mismo es o no una edición de bolsillo, por ejemplo, si el libro es “Programación en JAVA”, su título se modificará a “Programación en JAVA – De Bolsillo”
- Para las **revistas**, se agrega además el número de volumen (que se solicita mediante el método **getNroVolumen()** de la clase **Generador** que se descarga del entorno).

Se pide:

- Implementar la clase **Ejemplar** con sus respectivos atributos, constructores y métodos para acceder y modificar atributos.
- Implemente aquellas **clases, métodos y objetos** que considere necesarios para representar los diferentes tipos de ejemplares.
- Implementar la clase **ProgramaEditorial**, su método main y los métodos que considere necesarios para la lectura y almacenamiento de a lo sumo 1000 ejemplares. Luego, haga la impresión de un listado de todos los ejemplares existentes.
- Escriba todos los métodos que considere necesarios para publicar los ejemplares de la editorial (considere los atributos del estado interno que se verán modificados).

3- Una estación de servicio maneja las cargas de combustible de los vehículos que pasan por sus surtidores. De cada surtidor se registra: el número de surtidor, el importe total que facturó en el día, el DNI, nombre y apellido del playero que lo administró.

Existen diferentes tipos de surtidores:

- **Surtidor de Gasoil**, debido a la escasez del combustible, este surtidor posee un límite diario de clientes a los que puede garantizarles la carga. Por lo que, tiene asociado el conjunto de patentes de aquellos vehículos que pudieron cargar gasoil (a lo sumo 20).
- **Surtidor de Nafta**, este surtidor no tiene límites de clientes, por lo cual sólo registra la cantidad de litros que tiene para proveer durante el día.

Se pide:

- Implementar la clase **Surtidor** con sus respectivos atributos, constructores y métodos para acceder y modificar sus atributos. Además de todos los elementos necesarios para manejar los diferentes tipos de surtidores.
- La estación de servicio quiere saber cuánto dinero le queda de ganancia, para esto necesita calcular los impuestos que debe pagar y que se deducen de la siguiente manera:
  - En el caso de la **Nafta**, se retiene el 2% por cada litro que tenga disponible para cargar el surtidor.
  - En el caso del **Gasoil**, se realiza el siguiente cálculo: si se les cargó gasoil hasta 5 vehículos se le retiene el 5% del total facturado, si se les cargó a más de 5 y hasta 15 vehículos se les retiene el 3% del importe facturado y a más de 15 vehículos se le retiene el 1,5% del importe total.
- Implementar los métodos necesarios para imprimir toda la información de los surtidores.



- d. Implementar un programa principal que cargue dos **surtidores de gasoil** (uno con 3 vehículos y uno con 8 vehículos) y un **surtidor de nafta**. Después de la carga, aplicar las retenciones y hacer la impresión de los datos utilizando lo implementado en el inciso c).