

En una determinada empresa de estética, se aborda la mecanización de su control de almacén. En una primera entrevista con el gerente de la misma, se deducen una serie de requerimientos que se pretenden plasmar mediante clases de una librería **GeneralidadesEstetica**. Algunas de estas necesidades se describen a continuación:

**Manejo de albaranes de salida de material:** un albarán de salida refleja la entrega a un cliente de una o varias unidades de uno o varios artículos. En este proceso se quiere distinguir los siguientes conceptos/objetos:

- **Cabezas de albarán:** sirven para agrupar las características/funcionalidades genéricas implicadas en el proceso:
  - **Características/atributos**
    - Código de albarán: identificación del albarán, compuesto por:
      - Serie: cadena de 3 caracteres
      - Número: secuencial dentro de la serie de referencia.
    - Código de cliente: código de 4 dígitos que identifica al cliente.
    - Fecha de creación: refleja la fecha en que tiene lugar la transacción.
    - Estado del albarán: se pretende distinguir entre:
      - Creado
      - Enviado al cliente
      - Facturado
      - Cobrado
    - **Líneas de albarán:** serie de referencias a las líneas de albarán (descritas en este mismo dcto.) incluidas en el albarán en curso.
  - **Funcionalidades**
    - **Valorar:** debe devolver el importe correspondiente a la suma de importes de las líneas de albarán que incorpora.
    - **Serializar:** debe generar un fichero secuencial con un registro por cada línea de albarán que incorpora:
      - La ruta donde colocar el fichero se le suministrará mediante un parámetro.
      - El nombre de archivo será el código del albarán y extensión '.dat'
      - Debe devolver el objeto stream utilizado.
    - Cuando un objeto de este tipo se utilice acompañado de  $[n^o]$  debe devolver el objeto línea que ocupa la posición  $n^o$  en su colección de líneas. Un  $n^o$  inexistente generará una excepción de tipo `OverflowException`.
    - **SoyYo:** capaz de devolver la cabeza de albarán que recibe el mensaje cuando el parametro struct que mantiene el código (Serie y nº) coincide con los valores de los atributos correspondientes. NULL en caso contrario.
- **Líneas de albarán:** sirven para agrupar las características/funcionalidades implicadas en cada artículo manejado en la transacción:
  - **Características/atributos**
    - Código: identificación de la línea. Se compone de:
      - Código de albarán: identificación del albarán, compuesto por:
        - Serie: cadena de 3 caracteres
        - Número: secuencial dentro de la serie de referencia.
      - Nº de Línea: nº secuencial de la línea a nivel de albarán
    - Cabeza: objeto Cabeza de albarán al que pertenezca la línea
    - Artículo: objeto artículo correspondiente a la línea en curso.
    - Cantidad: la del artículo que se maneja
    - Precio unitario
    - Descuento aplicado
    - Estado de la línea: se pretende distinguir entre:
      - Anulada
      - Vigente
    -
  - **Funcionalidades**
    - Constructor: con un parámetro:
      - Objeto cabeza de albarán de pertenencia
    - debe
      - Enlazar el objeto receptor del mensaje con el albarán cuyo código se le suministre como parámetro:

- Actualiza su atributo Cabeza con el suministrado como parámetro.
  - Añade la línea actual a la colección de líneas de la cabeza de albarán que se le suministra como parámetro..
- **ActualizaLinea:** Partiendo de tres parámetros que representan:
  - Objeto artículo de referencia
  - cantidad y precio de venta
    - Debe:
      - Disminuir el stock y actualizar el precio medio de venta del artículo correspondiente al objeto línea en curso.
      - Actualizar su atributo Artículo con el suministrado como parámetro.
      - Actualizar su atributo Cantidad con la suministrada como parámetro
- Cuando se utilice el operador + entre dos objetos de este tipo, debe devolver un nuevo objeto de la misma clase donde:
  - El artículo sea el de cualquiera de los operandos.
  - La cantidad sea la suma de cantidades.
  - El precio unitario sea la media aritmética de precios.
  - Si ambos operandos no referencian al mismo código de artículo, se debe lanzar una excepción `FormatException`.
- **Artículos:** sirven para agrupar las características/funcionalidades de los artículos:
  - **Características/atributos**
    - Código de artículo: cadena de identificación del artículo
    - stock: nº de unidades en existencia.
    - Precio medio de venta
  - **Funcionalidades**
    - **Vender:** con dos parámetros: cantidad y precio de venta:
      - debe:
        - Actualizar el stock disminuyéndolo en la cantidad recibida.
        - Actualizar el precio medio de venta ponderando:
          - El stock actual al precio medio de venta actual con la cantidad y precios suministrados en el mensaje
  - Para comprobar su utilización se requiere crear una aplicación de consola que:
    - Cree un albarán con los valores necesarios suministrados por teclado.
    - Cree dos artículos con valores a elección del alumno.
    - Enganche 2 líneas de albarán con los datos que solicite por teclado.
    - Metodo capaz de devolver una tabla con los codigos de las lineas de albarán que correspondan a una serie de objetos cabezas de albarán que se le suministran, paramétricamente, como elementos de un vector (una fila de codigos de linea por cada elemento del vector de cabezas)