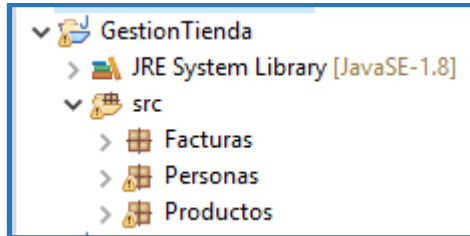


ALUMNO/A:

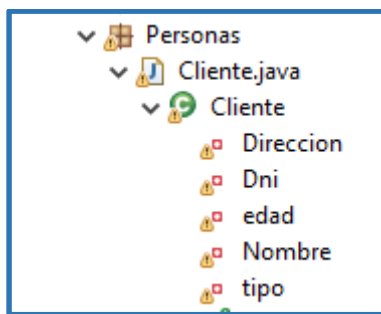
Crea un nuevo proyecto java llamado **GestionTienda** con la siguiente estructura de paquetes:



➤ **Paquete Personas: (1,5 puntos)**

Contiene la clase **Cliente**, que se estructura de la siguiente manera:

• **Atributos**



El **tipo** puede ser “normal” o “vip” (por defecto “normal”)

• **Constructores:**

- `public Cliente(dni, nombre)`
- `public Cliente(dni, nombre, direccion)`
- `public Cliente(dni, nombre, direccion, edad, tipo)`
- `public Cliente(Cliente c1)`

Pon valores por defecto en todos los atributos

• **Métodos:**

- Métodos getter y setter para cada atributo
- `public void mostrarCliente()`

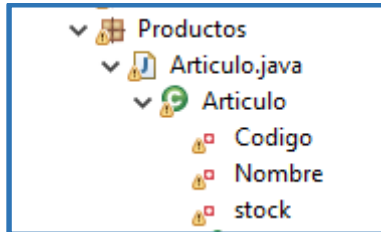
El método `mostrarCliente()` imprimirá todos los datos del cliente.

ALUMNO/A:

➤ **Paquete Productos: (1 punto)**

Contiene la clase **Articulo** que se estructura de la siguiente manera:

- **Atributos**



- **Constructores:**

- `public Articulo(codigo,nombre,stock)`

- **Métodos:**

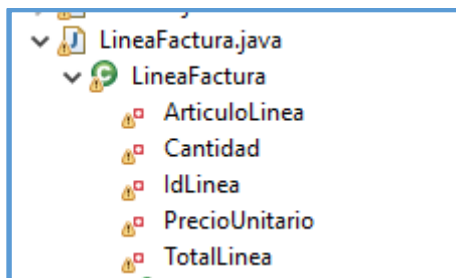
- Métodos getter y setter para cada atributo
- `public void mostrarArticulo()`

El método `mostrarArticulo()` imprimirá todos los datos del artículo.

➤ **Paquete Facturas:** Contiene las clases **LineaFactura** y **Factura**

Clase **LineaFactura**: **(2,5 puntos)**

- **Atributos**



- **ArticuloLinea:** Objeto de tipo **Articulo**, que indica el artículo que se vende.
- **Cantidad:** contiene la cantidad de unidades vendidas
- **IdLinea:** identificador del número de línea. Ej: **linea1**
- **PrecioUnitario:** Es el precio de venta de cada unidad del artículo
- **TotalLinea:** Precio total de la línea (Cantidad * precio unitario)

ALUMNO/A:

- **Constructor:**
 - `Public LineaFactura(idLinea, articuloLinea, precioUnitario, cantidad)`

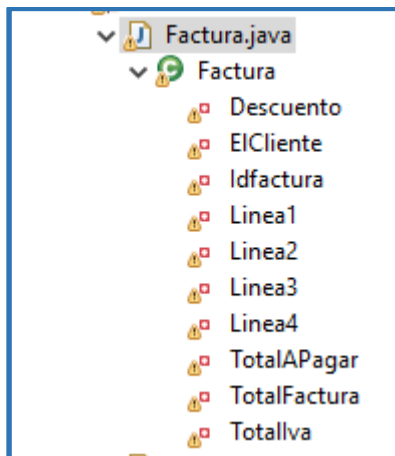
En el constructor debes tener en cuenta que:

- ✓ El stock del artículo se reducirá conforme a la cantidad vendida en la línea de la factura. Si se intenta vender una cantidad mayor a la disponible, la cantidad se asignará al stock disponible (que pasará a ser cero)
- **Métodos:**
 - Métodos getter y setter para cada atributo
 - `public void mostrarLinea()`

El método `mostrarLinea()` imprimirá todos los datos de la línea.

Clase **Factura**: **(3.5 puntos)**

- **Atributos**



- **IdFactura:** Identificador de la factura
- **ElCliente:** objeto de la clase Cliente, el cliente comprador
- **Linea1, Linea2, Linea3, Linea4:** 4 objetos de la clase LineaFactura, suponemos que todas las facturas tienen obligatoriamente 4 líneas de factura.
- **TotalIva:** IVA aplicado (21% aplicado a la suma de las 4 líneas de factura)
- **TotalFactura:** Precio total de la factura (la suma del precio de las 4 líneas de la factura más el iva)
- **Descuento:** Si el cliente es VIP se aplica el 10% al precio total de la factura. En caso contrario el descuento será cero.
- **TotalApagar:** Precio total de la factura menos el descuento.

ALUMNO/A:

- **Constructor:**
 - `public Factura(idfactura, elcliente, linea1, linea2, linea3, linea4);`

El constructor asigna los valores apropiados a los atributos conforme a lo indicado anteriormente.

- **Métodos:**
 - Métodos getter y setter para cada atributo
 - `public void mostrarFactura()`

El método `mostrarFactura()` imprimirá todos los datos de la factura de la siguiente manera:

```
FACTURA: 78M10
*****
*      78787845T
*      jose Luis
*      C/Mayor
*
*****

CÓDIGO  NOMBRE  N.UNIDADES  PRECIO  IMPORTE
Rat564  Raton    3           5.0     15.0
Tec495  Teclado   1           15.0    15.0
Mon307  Monitor   2          150.0   300.0
Dis833  Discos    5           20.0   100.0

TOTAL SIN IVA: 430.0
IVA 21%: 90.3
-52.03
*****
TOTAL
468.27
*****

Por ser VIP te descontamos 10%
```

Nota: Para imprimir así las líneas, el cliente y el total, usa el carácter `\t` que es un tabulador por ejemplo: `codigo+"\t"+nombre+"\t\t"+nunidades...`

Programa principal: (1.5 puntos)

Implementa un programa principal con una factura similar a la que se muestra en el ejemplo. Utiliza la función `mostrarFactura()` para mostrar la factura. Incluye comentarios para facilitar la comprensión de tu programa.