Sobrecargas

Se pide resolver cada uno de los ejercicios utilizando el lenguaje de programación Java. En todos los casos, reutilizar código.

[D.01] - Qué cosa, la cosa...

Diseñar la clase Cosa con tres atributos privados:

- entero (para almacenar valores numéricos enteros)
- cadena (para guardar valores de tipo cadena de caracteres)
- fecha (para poder almacenar valores de fechas)

El método establecerValor, tendrá tres sobrecargas, una para cada atributo de la clase. No retornarán ningún valor.
El método (de instancia) mostrar() no recibirá parámetros y retornará, en formato de cadena, los valores de cada uno de los atributos, separados por guiones medios.
El método de clase mostrar(Cosa), recibe cómo parámetro una instancia de la clase Cosa y retornará, en formato de cadena, los valores de cada uno de los atributos, separados por guiones medios.

En el método main, probar todos los miembros de la clase Cosa.

[D.02] - Tinta y pluma

Hay que modelar la clase Tinta que posee dos atributos privados, constructor sobrecargado y métodos sobrecargados.
Atributos:

- color (de tipo Color que es un enumerado que posee cómo enumeraciones: *Rojo*, *Blanco*, *Azul*, *Verde* y *Negro*)
- tipo (otro enumerado con *Común*, *China* y *ConBrillito*)

Constructor (con sobrecargas)

- que posea cero parámetros.
- que posea un parámetro.
- que posea dos parámetros.

Nota: *en los casos dónde no se reciban todos los parámetros, los valores predeterminados serán: Verde y China.*

Método (de instancia)

- mostrar(): String. Método privado que retorna en formato de cadena los valores de cada atributo del objeto. Utilizar la clase StringBuilder para armar la cadena de texto.

Métodos (de clase)

- mostrar(Tinta): String. Método público que recibe un parámetro de tipo Tinta y retorna en formato de cadena los valores de cada atributo del objeto.
- sonIguales(Tinta, Tinta): Boolean. Recibe dos parámetros de tipo Tinta y retorna ***true***, si el valor del color y del tipo son iguales en ambos objetos.
- sonDistintos(Tinta, Tinta): Boolean. Recibe dos parámetros de tipo Tinta y retorna ***false***, si el valor del color y del tipo son iguales en ambos objetos.

La siguiente clase que hay que modelar es la clase Pluma, la cual posee tres atributos privados, constructor (sobrecargado) y métodos sobrecargados.
Atributos:

- marca (de tipo cadena de caracteres)
- tinta (de tipo Tinta)
- cantidad (valor de tipo entero)

Constructor (con sobrecargas)

- que posea cero parámetros.
- que posea un parámetro.
- que posea dos parámetros.
- que posea tres parámetros.

Nota: *en los casos dónde no se reciban todos los parámetros, los valores predeterminados serán: "sin marca", null y 1, respectivamente.*

Método (de instancia)

- mostrar(): String. Método público que retorna en formato de cadena los valores de cada atributo del objeto. Utilizar la clase StringBuilder para armar la cadena de texto.

Métodos (de clase)

- sonIguales(Pluma, Tinta): boolean. Recibe dos parámetros, uno de tipo Pluma y otro de tipo Tinta. Retorna ***true***, si el valor del atributo tinta del primer parámetro es igual al segundo parámetro.
- sonDistintos(Pluma, Tinta): boolean. Recibe dos parámetros, uno de tipo Pluma y otro de tipo Tinta. Retorna ***false***, si el valor del atributo tinta del primer parámetro es igual al segundo parámetro.
- add(Pluma, Tinta): Pluma. Solo si el valor del atributo tinta (primer parámetro) es igual al segundo parámetro, se incrementa el valor del atributo cantidad en una unidad.
- remove(Pluma, Tinta): Pluma. Solo si el valor del atributo tinta (primer parámetro) es igual al segundo parámetro, se decrementa el valor del atributo cantidad en una unidad.

Nota: *No puede quedar una cantidad negativa.*

Generar un proyecto de tipo librería de clases, incorporar las clases Tinta y Pluma y en un proyecto de tipo consola, probar el correcto funcionamiento de los miembros de las clases modeladas.