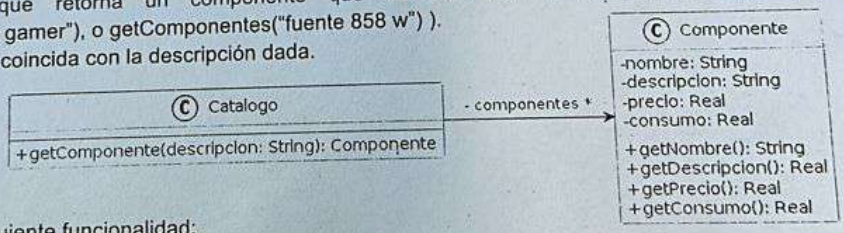
Ejercicio 1 - Patrones

OO2 -1er recuperatorio- 29/06/2024

Una empresa de venta de computadoras ofrece una variedad de configuraciones para satisfacer las necesidades de sus clientes. En la actualidad, la empresa ofrece tres configuraciones: básica, intermedia y gamer.
Cuando se solicita un presupuesto para un equipo, se registra también el nombre de la persona que hizo la solicitud y la fecha en que se realizó. Cabe destacar que cada presupuesto debe ser para un solo equipo.
Las configuraciones ofrecidas en este momento se muestran en la siguiente tabla:

	Basico	Intermedio	Gamer
Procesador	Procesador Básico	Procesador Intermedio	Procesador Gamer. Hay que agregar un pad térmico y un cooler
Memoria ram	8 GB	16 GB	32 gb + 32 gb
Disco	HDD 500 GB	SSD 500 GB	SSD 500gb + SSD 1 TB
Tarjeta gráfica	No posee (integrada)	GTX 1650	RTX 4090.
Gabinete	Gabinete Estándar (ya viene con fuente)	Gabinete Intermedio. Fuente 800 w	Gabinete Gamer Para saber que fuente requiere, se debe sumar el consumo de sus componentes + 50% de ese consumo. Luego ese resultado debe ser incluido en la descripción de la forma "fuente consumo w".

Para simplificar la atención a sus clientes, no ofrece componentes sueltos, sólo equipos definidos por sus técnicos. En el futuro, la empresa está interesada en ampliar constantemente su oferta mediante la incorporación de nuevas configuraciones de equipos.
Para resolverlo, se cuenta con una clase **Catálogo ya implementada** que ofrece un método *#getComponente(String)* que retorna un componente que coincide con la descripción dada (ej, *getComponentes*("gabinete gamer"), o getComponentes("fuente 858 w")).
Siempre retornará uno que coincida con la descripción dada.



Ud debe implementar la siguiente funcionalidad:

- **Crear presupuestos** para las configuraciones mostradas. Tenga en cuenta que su solución debe facilitar el lanzamiento de nuevas configuraciones.
- **Calcular el consumo de un equipo:** El consumo de un equipo está formado por la suma de los consumos de cada uno de sus componentes.
- **Calcular el precio de un equipo:** El precio final de un equipo está formado por la suma de los precios de cada uno de sus componentes más el 21% de IVA.

Tareas:

1. Modele una solución usando un diagrama UML para el problema planteado utilizando alguno de los patrones vistos en la materia. Indique cuáles y los roles en su diseño.
2. Implemente en Java la funcionalidad requerida.
3. Liste los pasos necesarios, de forma breve, los cambios que deben realizarse en su solución si se tiene la necesidad de agregar nuevas configuraciones. Especifique si se deben agregar subclases, métodos en clases existentes, renombrar métodos, etc.
4. La empresa tiene la intención de incorporar otras configuraciones que agregan monitores y periféricos. ¿Qué cambios debería realizar en su solución? Liste los pasos necesarios para hacerlo (especifique si se deben agregar subclases, métodos en clases existentes, renombrar métodos, etc).

Ejercicio 2 - Refactoring

OO2 -1er recuperatorio- 29/06/2024

Para el siguiente código, realice las siguientes tareas:
(i) indique que mal olor presenta
(ii) indique el refactoring que lo corrige
(iii) aplique el refactoring (modifique el código)
Si vuelve a encontrar un mal olor, retorne al paso (i).

Nota: Haga los cambios que considere necesarios.

```
1. public class Pago {
3.    private List<Producto> productos;
4.    private String tipo;
5.    private static final double ADICIONAL_TARJETA = 1000.0;
6.    private static final double DESCUENTO_EFECTIVO = 2000.0;

7.    public Pago(String tipo, List<Producto> productos) {
8.       this.productos = productos;
9.       this.tipo = tipo;
10.   }

12.   public double calcularMontoFinal() {
13.      double total = 0.0;
14.      if (this.tipo == "EFECTIVO"){
15.         for (Producto producto: this.productos){
16.            total = total + producto.getPrecio() + (producto.getPrecio() * producto.getIVA());
17.         }
18.         if (total > 100000){
19.            total = total - DESCUENTO_EFECTIVO;
20.         }
21.      }
22.      else if (this.tipo == "TARJETA"){
23.         for (Producto producto: this.productos){
24.            total = total + producto.getPrecio() + (producto.getPrecio() * producto.getIVA())
25.         }
26.         total = total + ADICIONAL_TARJETA;
27.      }
28.      return total;
29.   }
30. }
```

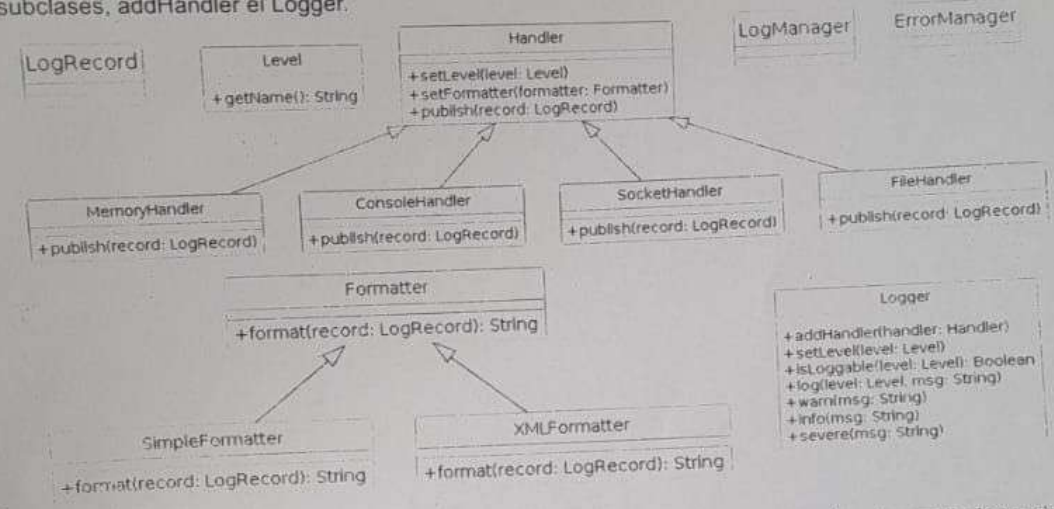
```
1.  public class Producto {
2.     private double precio;
3.     private double IVA;
4.
5.     public Producto(double precio, double IVA) {
6.        this.precio = precio;
7.        this.IVA = IVA;
8.     }
9.
10.    public double getPrecio() {
11.       return this.precio;
12.    }
```

```
13.   public double getIVA() {
14.         return this.IVA;
15.   }
16. }
```

Ejercicio 3 - Frameworks

OO2 -1er recuperatorio- 29/06/2024

Consideremos el framework de Logging visto en la materia. El framework permite loguear eventos importantes, errores, etc. Estos logs son útiles para desarrolladores, administradores y usuarios. El framework permite enviar mensajes de log a consola, archivos, sockets, etc.
Como ayuda memoria se ofrece un diagrama simplificado de algunas clases del framework. Note algunos métodos importantes como publish en handler y sus subclases, format en formatter y sus subclases, addHandler el Logger.



El framework se encarga de crear y gestionar las instancias de Loggers que los desarrolladores de una aplicación utilizarán para loguear. La configuración de los loggers generalmente se realiza al iniciar la aplicación (y se mantiene globalmente). Esto significa que se pueden establecer niveles de log, formatos de salida, destinos de los logs y otras opciones relacionadas con la generación y gestión de logs.

1. Se está desarrollando una aplicación que requiere usar este framework de la siguiente manera: Se requiere tener dos loggers, uno para loguear en consola (salida a pantalla) y otro para loguear en un archivo con formato XML. Explique brevemente cómo utilizaría el framework para implementar esta funcionalidad (cómo lo configuraría, que clases provistas utilizaría, qué clases se deben extender, qué métodos se deben agregan o implementar, etc).

2. En otra aplicación se requiere utilizar el framework para utilizar un logger que preste atención a los eventos a partir del nivel SEVERE y enviarlos por WhatsApp. Explique brevemente cómo utilizaría el framework para implementar esta funcionalidad (cómo lo configuraría, que clases provistas utilizaría, qué clases se deben extender, qué métodos se deben agregan o implementar, etc). Para enviar un mensaje por whatsapp escriba la siguiente expresión: Whatsapp.enviar(String mensaje, String teléfono).

3. Se necesita desarrollar una aplicación utilizando este framework para publicar los logs por mail. ¿Puede asegurar que en la implementación de esta funcionalidad el programador notará la existencia de inversión de control? Justifique su respuesta en términos de que debe hacer el programador y qué observa cuando el código se ejecuta.