

EVALUACION N°2 (ABP)

Plan Formativo	Nivel de Dificultad:
Full Stack Java	Medio
Nombre del proyecto: Electrodomésticos	Tema: Programación Orientada a Objetos
Objetivo del proyecto:	Evaluación programación orientada a objetos, se debe emplear código Java, basada en conceptos y buenas prácticas de la industria normada por la programación orientada a objetos.
Ejecución: Individual	
Descripción del proyecto	
<p>Crear una superclase llamada Electrodoméstico con las siguientes características:</p> <ul style="list-style-type: none"> Sus atributos son precio base, color, consumo energético (letras entre A y F) y peso. Indica que se podrán heredar. Por defecto, el color será blanco, el consumo energético será F, el precioBase es de \$100.000 y el peso de 5 kg. Usar constantes para ello. Los colores disponibles son blanco, negro, rojo, azul y gris. No importa si el nombre está en mayúsculas o en minúsculas. Los constructores que se implementarán serán <ul style="list-style-type: none"> Un constructor por defecto. Un constructor con el precio y peso. El resto por defecto. Un constructor con todos los atributos. Los métodos que implementara serán: <ul style="list-style-type: none"> Métodos get de todos los atributos. comprobarConsumoEnergetico(char letra): comprueba que la letra es correcta, sino es correcta usara la letra por defecto. Se invocará al crear el objeto y no sera visible. comprobarColor(String color): comprueba que el color es correcto, sino lo es usa el color por defecto. Se invocará al crear el objeto y no será visible. precioFinal(): según el consumo energético, aumentara su precio, y según su tamaño también. 	

Esta es la lista de precios:

ETRA	PRECIO
A	100 €
B	80 €
C	60 €
D	50 €
E	30 €
F	10 €
TAMAÑO	PRECIO
Entre 0 y 19 kg	10 €
Entre 20 y 49 kg	50 €
Entre 50 y 79 kg	80 €
Mayor que 80 kg	100 €

Crearemos una subclase llamada **Lavadora** con las siguientes características:

- Su atributo es **carga**, además de los atributos heredados.
- Por defecto, la carga es de 5 kg. Usa una constante para ello.
- Los constructores que se implementarán serán:
 - Un constructor por defecto.
 - Un constructor con el precio y peso. El resto por defecto.
 - Un constructor con la carga y el resto de atributos heredados. Recuerda que debes llamar al constructor de la clase padre.
- Los métodos que se implementara serán:
 - Método get de carga.
 - **precioFinal()**; si tiene una carga mayor de 30 kg, aumentara el precio 50 €, sino es así no se incrementara el precio. Llama al método padre y añade el código necesario. Recuerda que las condiciones que hemos visto en la clase Electrodoméstico también deben afectar al precio.

Crearemos una subclase llamada **Televisión** con las siguientes características:

- Sus atributos son **resolución** (en pulgadas) y **sintonizador TDT** (booleano), además de los atributos heredados.
- Por defecto, la resolución será de 20 pulgadas y el sintonizador será false.
- Los constructores que se implementarán serán:
 - Un constructor por defecto.

- Un constructor con el precio y peso. El resto por defecto.
- Un constructor con la resolución, sintonizador TDT y el resto de atributos heredados. Recuerda que debes llamar al constructor de la clase padre.
- Los métodos que se implementara serán:
 - Método get de resolución y sintonizador TDT.
 - **precioFinal()**: si tiene una resolución mayor de 40 pulgadas, se incrementara el precio un 30% y si tiene un sintonizador TDT incorporado, aumentara 50 €. Recuerda que las condiciones que hemos visto en la clase Electrodoméstico también deben afectar al precio.

Ahora crea una clase ejecutable que realice lo siguiente:

- Crea un array de Electrodomésticos de 10 posiciones.
- Asigna a cada posición un objeto de las clases anteriores con los valores que desees.
- Ahora, recorre este array y ejecuta el método `precioFinal()`.
- Deberás mostrar el precio de cada clase, es decir, el precio de todas las televisiones, por un lado, el de las lavadoras por otro y la suma de los Electrodomésticos (puedes crear objetos `Electrodoméstico`, pero recuerda que `Televisión` y `Lavadora` también son electrodomésticos). Recuerda el uso operador `instanceof`.

Por ejemplo, si tenemos un `Electrodoméstico` con un precio final de 300, una lavadora de 200 y una televisión de 500, el resultado final será de 1000 ($300+200+500$) para electrodomésticos, 200 para lavadora y 500 para televisión.

Contribuciones

Requerimientos de los participantes		
Conocimientos previos <ul style="list-style-type: none"> • Programación básica en Java. • Programación orientada a objetos. • Herencia y Polimorfismos • UML 	Actitudes para el trabajo <ul style="list-style-type: none"> • Cumplimiento de plazos • Buenas prácticas de codificación • Diseño y Estructura • Trabajo en equipo 	Valores <p>Tiempo de resolución.</p> <p>Enfoque al requerimiento.</p> <p>Estructura de Solución.</p>
Objetivo General de Aprendizaje	El participante al finalizar el proyecto será capaz de: <p>Hacer uso de la programación orientada a objetos en la resolución de un problema</p>	
Objetivos particulares	<ul style="list-style-type: none"> - Resolución del problema individualmente - Codificar pieza de código que resuelve problema planteado. - Utilización de programas de almacenamiento y versionamiento en línea - Codificación según las buenas practicas y estándares. - Diagramas de Clases 	

Duración del proyecto		Máximo 2 clases, con revisión en la última de las sesiones.		
Criterios de Evaluación				
Utiliza estructuras de control condicional y expresiones lógicas para dar solución a un problema de desarrollo.	Utiliza estructuras de control repetitivas para dar solución a un problema de repetitividad.	Hace uso de la programación orientada a objetos y sus conceptos base para la resolución de un problema. (Herencia, Polimorfismo)	Elabora un algoritmo que responda a una situación planteada de forma exacta.	Codifica de manera ordenada, incluyendo comentarios que permitan generar documentación de forma automática.
Tips o listado de Preguntas Guía				
Productos a obtener durante la realización del proyecto				
Proyecto Java desarrollado en base a clases y otros artefactos anexos. Diagrama de Clases de proyecto				
Especificaciones de desempeño				
Deberá realizar la actividad según requerimientos técnicos y en un plazo máximo de 2 clases, el resultado deberá ser entregado como un proyecto Java, acompañado por archivo plano Readme.txt contextualizando el problema y planteamiento de la solución, gestionado a repositorio GITHUB para su posterior revisión.				
Cronograma de actividades				
Sugerencias bibliográficas para la investigación				
Tutorial de Java <ul style="list-style-type: none">- https://docs.oracle.com/javase/tutorial/ Números aleatorios en Java <ul style="list-style-type: none">- http://lineadecodigo.com/java/numero-aleatorio-en-java/ Generar documentación con Javadoc <ul style="list-style-type: none">- https://www.youtube.com/watch?v=hwB4oFf_g4Q				