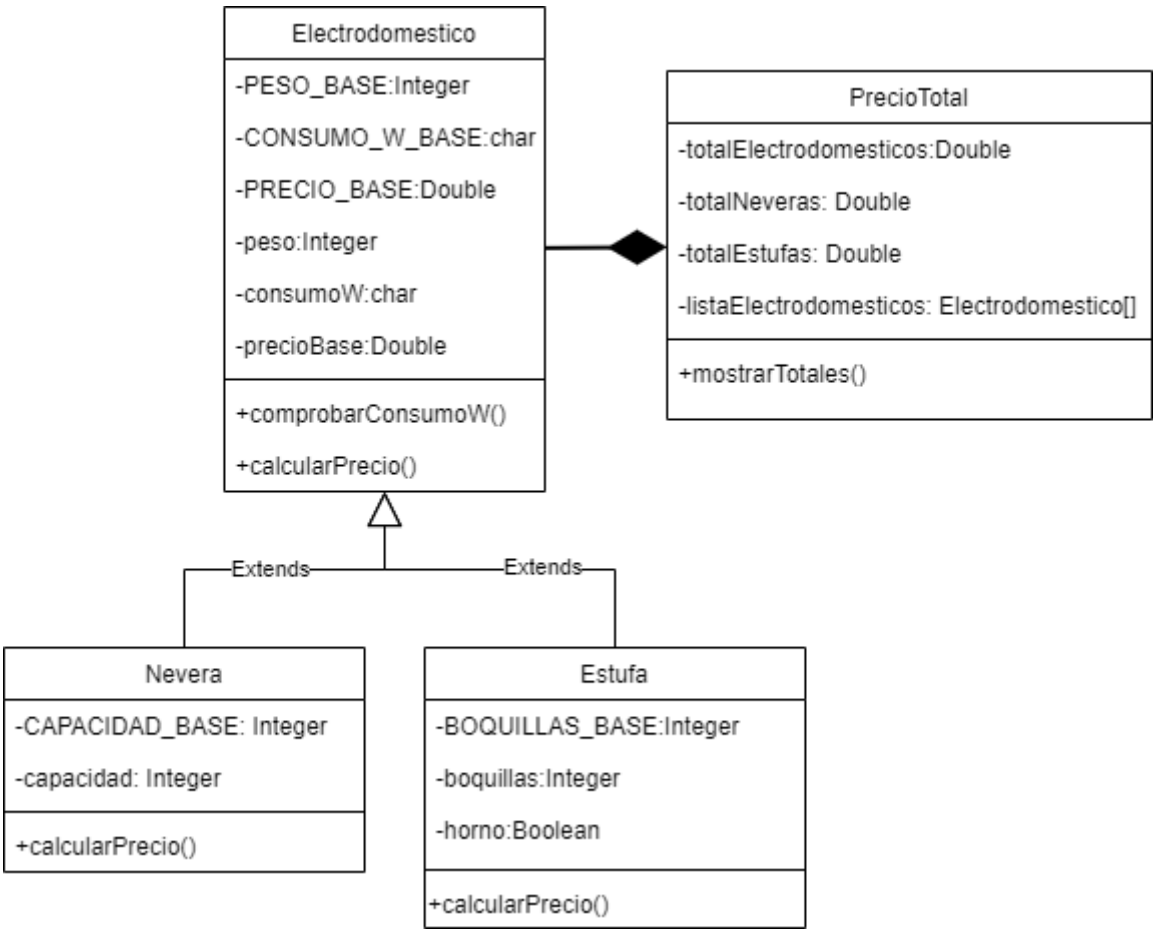
Ciclo 2 Fundamentos de programación

Reto 2-GRUPO51

Descripción del problema:

La cadena de almacenes “Grandes Superficies” hizo una compra de un lote de electrodomésticos, se pagó un único precio por todo el lote.
Se quiere determinar el valor de cada uno de los ítems en el mercado actual de acuerdo con unas condiciones y restricciones. Además del valor total de todos los productos.
Se contrató a usted como consultor para desarrollar la solución, en donde se determinó que el modelo de clases con el cual se resolverá el problema es el siguiente:



Todos los electrodomésticos comparten los atributos peso, consumoW, los cuales son valores que pueden modificar su precio final, se cuenta también con el atributo precioBase, el cual representa el precio del electrodoméstico antes de sumar los respectivos valores según las condiciones establecidas para los atributos peso, consumoW. El método calcularPrecio() permite obtener el precio final para cualquier electrodoméstico de la siguiente forma:

$$precioFinal = precioBase + adicion$$

La adición se establece según lo siguiente:

Electrodomestico

Si el consumo energético (consumoW) del electrodoméstico es ‘A’, se adiciona $200, si es ‘B’ se adiciona $150, si es ‘C’ se adiciona $125, si es ‘D’ se adiciona $100, si es ‘E’ se adiciona $75 y si es ‘F’ se adiciona $50.

Del mismo modo, si el peso del electrodoméstico es mayor o igual a 0 y menor que 15 se adiciona $20, si el peso es mayor o igual a 15 y menor a 30 se adiciona $40, si el peso es mayor o igual que 30 y menor a 50 se adiciona $70 y si el peso es mayor o igual a 50 se adiciona $100.

Nevera

Los electrodomésticos de la clase Nevera cuentan con el atributo capacidad(litros), el cual agrega un valor adicional al precio de la Nevera de la siguiente forma:

Si la capacidad de la Nevera es mayor a 250, se adiciona $100.

Por defecto, se tiene un valor de capacidad constante para las neveras:

CAPACIDAD_BASE = 100

Si no se tiene un valor para el atributo capacidad, este debe inicializarse con el valor de la constante.

Estufa

Los electrodomésticos de la clase Estufa cuentan con los atributos boquillas, horno los cuales agregan un valor adicional al precio de la estufa de la siguiente forma:

Si la estufa tiene horno se adiciona $150, si la estufa tiene mas de 2 boquillas se adiciona 40% sobre el precio base.

Por defecto, las estufas no tienen horno, Se tiene un valor constante para las boquillas:

BOQUILLAS_BASE = 2

Si no se tiene un valor para el atributo boquillas, se debe inicializar este con el valor de la constante.

Mas aspectos sobre Electrodomestico:

Cada uno de los atributos de la clase Electrodoméstico, tienen una constante que define un valor por defecto con el cual se puede calcular el precio del Electrodoméstico si no se envía ningún valor al constructor de la clase de la siguiente manera:

PESO_BASE = 10
CONSUMO_W_BASE = ‘F’
PRECIO_BASE = 50.0

Si no se tiene un valor para los atributos, se debe inicializar estos con el valor de las constantes.

PrecioTotal

Los atributos de la clase PrecioTotal son: totalElectrodomesticos, totalNeveras, totalEstufas y listaElectrodomesticos, por defecto sus valores son 0. El atributo listaElectrodomesticos contiene todos los electrodomésticos recibidos en la compra, los cuales son almacenados en un array(de tipo Electrodomestico), estos son instanciados en la clase App en su método main(), para ser entregados al constructor de la clase PrecioFinal.

En main(), además se llama al método mostrarTotales(), el cual debe imprimir en consola:

La suma del precio de los electrodomésticos es de {totalElectrodomesticos}
La suma del precio de las neveras es de {totalNeveras}
La suma del precio de las estufas es de {totalEstufas}

Ejemplo:

Pruebas	Salida

`Electrodomestico electrodomesticos1[]=new Electrodomestico[10];`
`electrodomesticos1[0]=new Electrodomestico(200.0, 60, 'C');`
`electrodomesticos1[1]=new Nevera(150.0, 30);`
`electrodomesticos1[2]=new Estufa(500.0, 80, 'E', 42, false);`
`electrodomesticos1[3]=new Electrodomestico();`
`electrodomesticos1[4]=new Electrodomestico(600.0, 20, 'D');`
`electrodomesticos1[5]=new Nevera(300.0, 40, 'Z', 40);`
`electrodomesticos1[6]=new Estufa(250.0, 70);`
`electrodomesticos1[7]=new Nevera(400.0, 100, 'A', 15);`
`electrodomesticos1[8]=new Estufa(200.0, 60, 'C', 30, true);`
`electrodomesticos1[9]=new Electrodomestico(50.0, 10);`
`PrecioTotal solucion1 = new PrecioTotal(electrodomesticos1);`
`solucion1.mostrarTotales();`
	La suma del precio de los Electrodomésticos es de 4725.0
	La suma del precio de las Neveras es de 1390.0
	La suma del precio de las Estufas es de 1930.0

`Electrodomestico electrodomesticos2[]=new Electrodomestico[5];`
`electrodomesticos2[0]=new Electrodomestico(300.0, 80, 'C');`
`electrodomesticos2[1]=new Nevera(150.0, 20);`
`electrodomesticos2[2]=new Estufa(500.0, 80, 'E', 42, false);`
`electrodomesticos2[3]=new Electrodomestico();`
`electrodomesticos2[4]=new Electrodomestico(600.0, 20, 'D');`
`PrecioTotal solucion2 = new PrecioTotal(electrodomesticos2);`
`solucion2.mostrarTotales();`
	La suma del precio de los Electrodomésticos es de 2500.0
	La suma del precio de las Neveras es de 240.0
	La suma del precio de las Estufas es de 875.0

NOTA: Las pruebas son ejecutadas en la clase App. Esta clase no se debe subir a la plataforma como parte de la solución.

Esqueleto:

```
// Inicio de la solución
public class PrecioTotal {
    // Atributos

    // Constructor
    PrecioTotal(Electrodomestico[] pElectrodomesticos) {
        //Codigo
    }

    //Metodos
    public void mostrarTotales() {
        // Código


        // Mostramos los resultados
        System.out.println("La suma del precio de los Electrodomésticos es de " + totalElectrodomesticos);
        System.out.println("La suma del precio de las Neveras es de " + totalNeveras);
        System.out.print("La suma del precio de las Estufas es de " + totalEstufas);
    }
}

public class Electrodomestico {
    // Constantes y Atributos


    //Constructores
    public Electrodomestico(){
        //Código
    }

    public Electrodomestico(Double precioBase, Integer peso){
        //Código
    }

    public Electrodomestico(Double precioBase, Integer peso, char consumoW){
        //Código
        comprobarconsumoW(consumoW);
    }

    // Metodos
    public void comprobarconsumoW(char consumoW){
        if(/* condicion */ ){
            this.consumoW=consumoW;
        }else{
            this.consumoW=CONSUMO_W;
        }
    }

    public Double calcularPrecio(){
        //Código

        return precioBase+adicion;
    }
}

public class Nevera extends Electrodomestico{
    // Constantes y Atributos
```

```
    //Constructores
    public Nevera(){
        //Código
    }
    public Nevera(Double precioBase, Integer peso){
        //Código
    }

    public Nevera(Double precioBase, Integer peso, char consumoW, Integer capacidad){
        //Código
    }

    // Métodos
    public Double calcularPrecio(){
        //Código
    }
}

public class Estufa extends Electrodomestico{
    // Constantes y Atributos

    //Constructores

    public Estufa(){
        //Código
    }
    public Estufa(Double precioBase, Integer peso){
        //Código
    }

    public Estufa(Double precioBase, Integer peso, char consumoW, Integer boquillas, boolean horno){
        //Código
    }

    // Métodos
    public Double calcularPrecio(){
        //Código
    }
}

//Fin de la Solución

// Esta clase es para las pruebas, no se debe subir como parte de la solución
public class App
{
    public static void main( String[] args )
    {
        Electrodomestico electrodomesticos2[]=new Electrodomestico[5];
        electrodomesticos2[0]=new Electrodomestico(300.0, 80, 'C');
        electrodomesticos2[1]=new Nevera(150.0, 20);
        electrodomesticos2[2]=new Estufa(500.0, 80, 'E', 42, false);
        electrodomesticos2[3]=new Electrodomestico();
        electrodomesticos2[4]=new Electrodomestico(600.0, 20, 'D');
        PrecioTotal solucion2 = new PrecioTotal(electrodomesticos2);
        solucion2.mostrarTotales();
    }
}
```

Nota: Recuerde que cada una de las clases debe ser codificada en una clase (archivo independiente), pero se deben de cargar juntas en iMaster.