

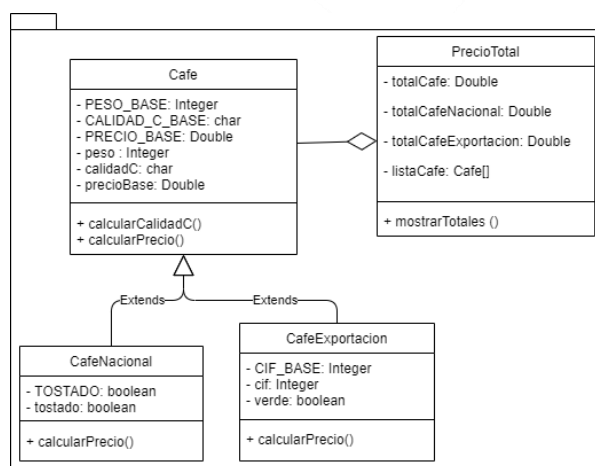


Ciclo 2 Fundamentos de programación

Reto 2

Descripción del problema:

La Trilladora de Café “Aroma de Mujer” necesita calcular el valor de sus ventas de café. Se desea determinar el valor total por cada uno de los tipos de café que maneja, así como el valor total por todo el café. Se te ha contratado para desarrollar una aplicación que dé solución al requerimiento a partir del siguiente diagrama:



Todos los tipos de café comparten los atributos peso y calidadC, los cuales son atributos que pueden modificar su precio final, se cuenta también con el atributo precioBase, el cual representa el precio del café antes de sumar las respectivas adiciones según las condiciones establecidas para los atributos peso, calidadC. El método calcularPrecio() permite obtener el precio final para cualquier tipo de café de la siguiente forma:

$$\text{precioFinal} = \text{precioBase} + \text{adicion}$$

La adición se establece según lo siguiente:

Si la calidad del café (calidadC) del café es ‘A’, se adiciona \$10, si es ‘B’ se adiciona \$8, si es ‘C’ se adiciona \$6, si es ‘D’ se adiciona \$5, si es ‘E’ se adiciona \$3 y si es ‘F’ se adiciona \$1.

Del mismo modo, si el peso del café es mayor o igual a 0 y menor que 19 se adiciona \$10, si el peso es mayor a 19 y menor a 49 se adiciona \$50, si el peso es mayor que 49 y menor a 80 se adiciona \$80 y si el peso es mayor a 79 se adiciona \$100

El café de la clase CafeNacional cuenta con el atributo tostado, el cual agrega un valor adicional al precio del café de la siguiente forma:

Si tostado es true, se adiciona \$50.

Por defecto, se tiene un valor de tostado constante para el CafeNacional:

TOSTADO = false

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

El café de la clase CafeExportacion cuenta con los atributos cif y verde los cuales agregan un valor adicional al precio del café tipo exportación de la siguiente forma:

Si el café tipo exportación es verde se adiciona \$50, si el CafeExportacion es de más de 40 cif se adiciona el 30% sobre el precio base.



Por defecto, el café tipo exportación no es verde. Se tiene un valor constante para el cif:

CIF_BASE = 20

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

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

CALIDAD_C_BASE = 'F'

PRECIO_BASE = 100.0

PESO_BASE = 5

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

Los atributos de la clase PrecioTotal son: totalCafe, totalCafeExportacion, totalCafeNacional y listaCafe, por defecto los valores de sus atributos son 0 o vacío, según sea el caso. El atributo listaCafe contiene todas las instancias de café vendido, las cuales son almacenados en un array (tipo Cafe), estas instancias se crean en la clase App en su método main(), para ser entregados al constructor de la clase PrecioTotal.

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

La suma del precio del café es de {totalCafe}

La suma del precio del café tipo nacional es de {totalCafeNacional}

La suma del precio del café tipo exportación es de {totalCafeExportacion}

Ejemplo:

Prueba	Salida
<pre>Cafe cafe[]=new Cafe[5]; cafe[0]=new Cafe(250.0, 50, 'D'); cafe[1]=new CafeNacional(180.0, 30); cafe[2]=new CafeExportacion(550.0, 80, 'B', 42, false); cafe[3]=new Cafe(); cafe[4]=new Cafe(550.0, 20, 'D'); PrecioTotal solucion1 = new PrecioTotal(cafe); solucion1.mostrarTotales(); System.out.println();</pre>	<p>La suma del precio del café es de 2105.0</p> <p>La suma del precio del café tipo nacional es de 231.0</p> <p>La suma del precio del café tipo exportación es de 823.0</p>
<pre>Cafe cafe2[]=new Cafe[10]; cafe2[0]=new Cafe(250.0, 50, 'D'); cafe2[1]=new CafeNacional(180.0, 30); cafe2[2]=new CafeExportacion(550.0, 80, 'B', 42, false); cafe2[3]=new Cafe(); cafe2[4]=new Cafe(550.0, 20, 'D'); cafe2[5]=new CafeNacional(300.0, 40, 'c', true); cafe2[6]=new CafeExportacion(250.0, 70); cafe2[7]=new CafeNacional(400.0, 100, 'B', true); cafe2[8]=new CafeExportacion(250.0, 50, 'D', 30, true); cafe2[9]=new Cafe(50.0, 10); PrecioTotal solucion2 = new PrecioTotal(cafe2); solucion2.mostrarTotales();</pre>	<p>La suma del precio del café es de 3841.0</p> <p>La suma del precio del café tipo nacional es de 1190.0</p> <p>La suma del precio del café tipo exportación es de 1539.0</p>



Esqueleto:

Para trabajar con el siguiente esquema su archivo .java debe nombrarse App.java

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

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

    // Constructor
    PrecioTotal(Cafe[] pElectrodomesticos) {
        // Código
    }

    public void mostrarTotales() {
        // Código
    }

    // Mostramos los resultados
    System.out.println("La suma del precio del café es de " + totalCafe);
    System.out.println("La suma del precio del café tipo nacional es de " + totalCafeNacional);
    System.out.print("La suma del precio del café tipo exportación es de " + totalCafeExportacio
n);
    }
}

public class Cafe {
    // Constantes y Atributos

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

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

    public Cafe(Double precioBase, Integer peso, char calidadC){
        // Código
        comprobarCalidadC(calidadC);
    }

    // Metodos
    public void comprobarCalidadC(char calidadC){
        if{// condición}{
            this.calidadC=calidadC;
        }else{
            this.calidadC=CALIDAD_C;
        }
    }

    public Double calcularPrecio(){
        // Código
        return precioBase + adicion;
    }
}

public class CafeNacional extends Cafe{
    // Constantes y Atributos

    //Constructor
    public CafeNacional(){
        // Código
    }

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



```
public CafeNacional(Double precioBase, Integer peso, char calidadC, boolean tostado){
    // Código
}

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

public class CafeExportacion extends Cafe{
    // Constantes y Atributos

    //Constructor
    public CafeExportacion(){
        // Código
    }

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

    public CafeExportacion(Double precioBase, Integer peso, char calidadC, Integer cif, boolean verda
e){
    // 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) {
        // Prueba 1
        Cafe cafe[]=new Cafe[5];
        cafe[0]=new Cafe(250.0, 50, 'D');
        cafe[1]=new CafeNacional(180.0, 30);
        cafe[2]=new CafeExportacion(550.0, 80, 'B', 42, false);
        cafe[3]=new Cafe();
        cafe[4]=new Cafe(550.0, 20, 'D');
        PrecioTotal solucion1 = new PrecioTotal(cafe);
        solucion1.mostrarTotales();
    }
}
```