

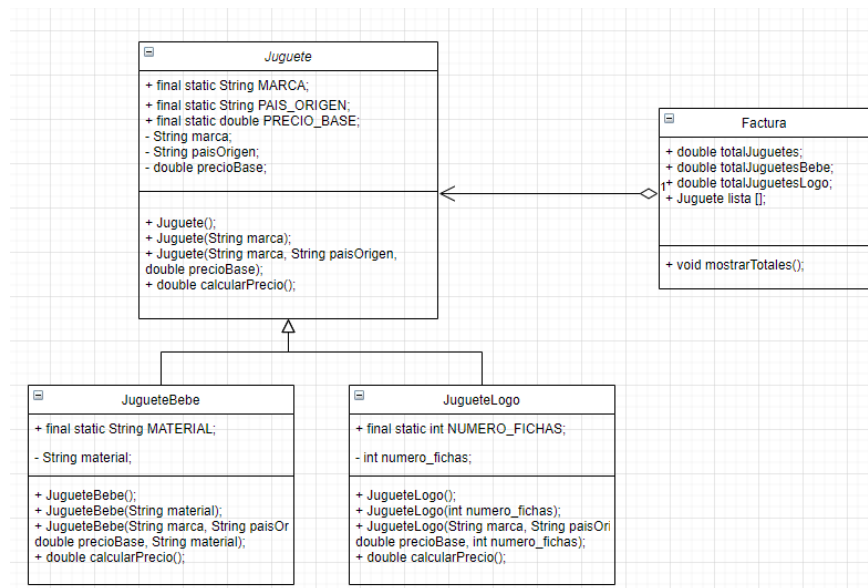


Ciclo 2 Fundamentos de programación

Reto 2 – Grupo 20

Descripción del problema:

La Fundación Miraflores recibirá en donación algunos juguetes. Necesita determinar el valor total por cada uno de los tipos de juguetes y el total para todos los juguetes donados. Para esto se ha contratado su compañía, en donde se ha determinado que el modelo de clases mediante el cual se resolverá el problema es el siguiente:



Todos los juguetes comparten los atributos marca y país de manufactura, los cuales son valores que pueden modificar su precio final, se cuenta también con el atributo **precioBase**, el cual representa el precio del juguete antes de sumar los respectivos valores según la evaluación de los atributos marca y país de manufactura. El método **calcularPrecio()** permite obtener el precio final para un **Jugete** Genérico de la siguiente forma:

Si la marca es Hasbro se disminuye el **precioBase** del juego en 50%, en caso contrario el **precioBase** disminuye en 25%. Luego de aplicado este descuento, si el país de manufactura es China se disminuye el valor del juguete en 5000 pesos, en caso contrario se disminuye en 2500. Nota: En caso de que los juguetes tengan un costo menor a 5000 o 2500 respectivamente no se hará el descuento.

Los juguetes del tipo **JugeteBebe** cuenta con el atributo **material**, el cual agrega un valor adicional al precio del ítem de la siguiente forma:

Si el material es plástico aumenta el precio del juguete en 10000 pesos, en caso contrario no hay aumento del precio

Los juguetes del tipo **JugeteLogo** cuentan con el atributo **numero_fichas**, el cual agrega un valor adicional al precio del ítem de la siguiente forma:

Si el **numero_fichas** es menor a 100 el juguete tiene un costo agregado de 15000 pesos, en caso contrario tiene un costo agregado de 20000.



Cada uno de los atributos de la clase `Juguete`, tienen una constante que define un valor por defecto con el cual se puede calcular el precio del juguete si no se envía ningún valor al constructor de la clase. Debe implementar 3 constructores.

Los atributos de la clase `Factura` son: `totalJuguetes`, `totalJuguetesBebe`, `totalJuguetesLego` y `listaJuguetes`. Este último atributo contiene todos los juguetes recibidos en donación, los cuales son almacenados en un array (tipo `Juguete`) y son entregados al constructor de la clase `Factura` en el método `main()`, desde donde se llama al método `mostrarTotales()`, el cual debe imprimir en consola:

El precio total de los juguetes es de {totalJuguetes}

La suma del precio de los `JuguetesBebe` es de {totalJuguetesBebe}

La suma del precio de los `JuguetesLego` es de {totalJuguetesLego}

Ejemplo:

Pruebas	Salida
<pre>Juguete juguetes[]=new Juguete[5]; juguetes[0]=new Juguete("Hasbro", "China", 30000); juguetes[1]=new JugueteBebe("Barbie", "Colombia", 75000, "Plastico"); juguetes[2]=new JugueteLego(500); juguetes[3]=new Juguete(); juguetes[4]=new JugueteBebe("Plastilina"); Factura solucion = new Factura(juguetes); solucion.mostrarTotales();</pre>	<pre>El precio total de los juguetes es de 108750.0 La suma del precio de los JuguetesBebe es de 68750.0 La suma del precio de los JuguetesLego es de 25000.0</pre>

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 Factura{
    // Atributos

    // Constructores

    // Metodos
    public void mostrarTotales(){
        // Calculo de totales

        System.out.println("El precio total de los juguetes es de " + totalJuguetes);
        System.out.println("La suma del precio de los JuguetesBebe es de " + totalJuguetesBebe);
        System.out.print("La suma del precio de los JuguetesLego es de " + totalJuguetesLego);
    }
}

public class Juguete{
    // Constantes
    private final static String MARCA="Hasbro";
    private final static String PAIS_ORIGEN="China";
    private final static String double PRECIO_BASE=10000;

    // Atributos
    private String marca;
    private String paisOrigen;
    private Double precioBase;
```



```
// Constructores

// Metodos
private tipoDato metodo1(){
    // En caso de ser necesarios metodos adicionales
}

public double calcularPrecio(){
    // Calculos
    return precioFinal;
}

// getters/setters de ser necesarios
}

public class JugueteBebe extends Juguete{
    // Constantes
    private final static String MATERIAL="Plastico";

    // Atributos
    private String material;
    // Constructores

    // Metodos
    public double calcularPrecio(){
        // Calculos
        return precioFinal;
    }

    // getters/setters de ser necesarios
}

public class JugueteLogo extends Juguete{
    // Constantes
    private final static int NUMERO_FICHAS = 100;
    // Atributos
    private int numero_fichas;
    // Constructores

    // Metodos
    public double calcularPrecio(){
        // Calculos
        return precioFinal;
    }

    // getters/setters de ser necesarios
}

// 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) {
        // Caso de Prueba 1
        Juguete juguetes[]=new Juguete[5];
        juguetes[0]=new Juguete("Hasbro", "China", 30000);
        juguetes[1]=new JugueteBebe("Barbie", "Colombia", 75000, "Plastico");
        juguetes[2]=new JugueteLogo(500);
        juguetes[3]=new Juguete();
        juguetes[4]=new JugueteBebe("Plastilina");

        Factura solucion = new Factura(juguetes);
        solucion.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.