

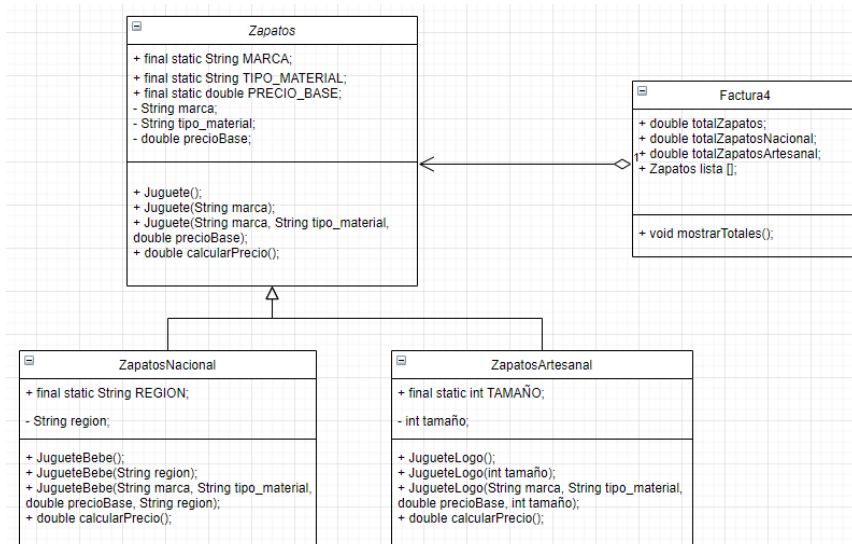


## Ciclo 2 Fundamentos de programación

### Reto 2 – Grupo 54

#### Descripción del problema:

La Fundación Amores recibirá en donación algunos zapatos. Necesita determinar el valor total por cada uno de los tipos de zapatos y el total para todos los zapatos 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 zapatos comparten los atributos `marca` y `tipo_material`, los cuales son valores que pueden modificar su precio final, se cuenta también con el atributo `precioBase`, el cual representa el precio del zapato antes de sumar los respectivos valores según la evaluación de los atributos `marca` y `tipo_material`. El método `calcularPrecio()` permite obtener el precio final para un Zapato Genérico de la siguiente forma:

Si la `marca` es Arturo se disminuye el `precioBase` del zapato en 75%, en caso contrario el `precioBase` disminuye en 20%. Si el `tipo_material` es Cuero se disminuye el valor del zapato en 5000 pesos, en caso contrario se disminuye en 500. Nota: En caso de que los zapatos tengan un costo menor a 5000 o 500 respectivamente no se hará el descuento.

Los zapatos del tipo **ZapatoNacional** cuentan con el atributo `región`, el cual agrega un valor adicional al precio del zapato de la siguiente forma:

Si la `región` es Cundinamarca aumenta el precio del zapato en 10000 pesos, en caso contrario no hay aumento del precio

Los zapatos del tipo **ZapatoArtesanal** cuentan con el atributo `tamaño`, el cual agrega un valor adicional al precio del zapato de la siguiente forma:

Si el `tamaño` es menor a 40 el zapato tiene un costo agregado de 10000 pesos, en caso contrario tiene un costo agregado de 4500 pesos.

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



Los atributos de la clase Factura4 son: totalZapatos, totalZapatosNacional, totalZapatosArtesanal, listaZapatos. Este último atributo contiene todos los zapatos recibidos en donación, los cuales son almacenados en un array (tipo Zapatos) y son entregados al constructor de la clase Factura4 en el método main(), desde donde se llama al método mostrarTotales(), el cual debe imprimir en consola:

El precio total de los zapatos es de { totalMedicamentos }

La suma del precio de los ZapatosNacionales es de { totalZapatosNacional }

La suma del precio de los ZapatosArtesanales es de { totalZapatosArtesanal }

### Ejemplo:

Pruebas	Salida
<pre>Zapatos zapatos[] = new Zapatos[5]; zapatos[0] = new Zapatos("Arturo", "Cuero", 5000); zapatos[1] = new ZapatosNacional("Valle"); zapatos[2] = new ZapatosArtesanal(42); zapatos[3] = new Zapatos(); zapatos[4] = new ZapatosNacional("Cundina marca");  Factura4 solucion = new Factura4(zapatos); solucion.mostrarTotales();</pre>	<pre>El precio total de los zapatos es de 68250.0 La suma del precio de los ZapatosNacional es de 35000.0 La suma del precio de los ZapatosArtesanales es de 17000.0</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 Factura4{
    // Atributos

    // Constructores

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

        System.out.println("El precio total de los zapatos es de " + totalZapatos);
        System.out.println("La suma del precio de los ZapatosNacional
es de " + totalZapatosNacional);
        System.out.print("La suma del precio de los
ZapatosArtesanal es de " + totalZapatosArtesanal);
    }
}

public class Zapatos{
    // Constantes
    private final static String MARCA="Arturo";
    private final static String TIPO_MATERIAL ="Cuerto";
    private final static String double PRECIO_BASE=10000;

    // Atributos
    private String marca;
    private String tipo_material;
    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 ZapatosNacional extends Zapatos {
    // Constantes
    private final static String REGION="Cundinamarca";

    // Atributos
    private String region;
    // Constructores

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

    // getters/setters de ser necesarios
}

public class ZapatosArtesanal extends Zapatos {
    // Constantes
    private final static int TAMAÑO = 40;
    // Atributos
    private int tamaño;
    // 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
        Zapatos zapatos[]=new Zapatos[5];
        zapatos[0]=new Zapatos("Arturo", "Cuero", 5000);
        zapatos[1]=new ZapatosNacional("Valle");
        zapatos[2]=new ZapatosArtesanal(42);
        zapatos[3]=new Zapatos();
        zapatos[4]=new ZapatosNacional("Cundinamarca");

        Factura4 solucion = new Factura4(zapatos);
        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.