

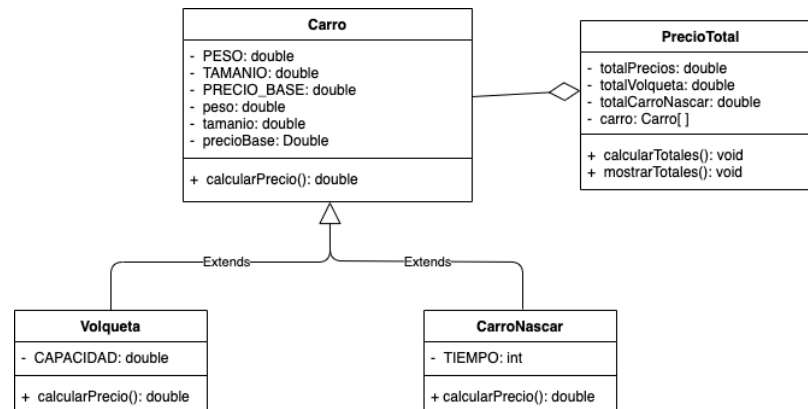


Ciclo 2 Fundamentos de programación

Reto 2

Descripción del problema:

Una empresa que brinda el servicio de fabricación de carros. Necesita determinar el valor total por cada uno de los tipos de carro y el total de todos los carros fabricados. 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 Carros comparten los atributos *peso* y *tamaño*, los cuales son valores que pueden modificar su precio final, se cuenta también con el atributo *precioBase*, el cual representa el precio del Carro antes de operar los respectivos valores según la evaluación de los atributos *peso* y *tamaño*. El método `calcularPrecio()` permite obtener el precio final para un tipo de Carro.

Los Carros de tipo *Volqueta* cuentan con la constante *CAPACIDAD*, la cual agrega un valor adicional al precio del Carro de la siguiente forma:

$\text{calcularPrecioCarro} = \text{precioBase} + (\text{peso} * \text{tamaño} * \text{CAPACIDAD})$.

Los Carros de tipo *CarroNascar* cuentan con la constante *TIEMPO*, la cual agrega un valor adicional al precio del Carro de la siguiente forma:

$\text{calcularPrecioCarroNascar} = \text{precioBase} + (\text{peso} * \text{tamaño} * \text{TIEMPO})$

Cada uno de los atributos de la super clase Carros, tienen una constante que define un valor por defecto a dichos atributos de la clase con los cuales se puede calcular el precio del Carro si no se envía ningún valor al constructor de la clase. Debe implementar 3 constructores (el primer constructor recibe el peso y tamaño, el segundo constructor recibe el precioBase y el tercer constructor no recibe ningún parámetro).

Los atributos de la clase *PrecioTotal* son: *totalPrecios*, *totalVolqueta*, *totalCarroNascar* y *carro*. Este último atributo contiene todos los Carros recibidos en la empresa, los cuales son almacenados en un array (tipo *Carro*) y son entregados al constructor de la clase *PrecioTotal* en el método `main()`, desde donde se llama al método `mostrarTotales()`, el cual debe imprimir en consola:

```
Total Carros {totalPrecios}
Total Volqueta {totalVolqueta}
Total CarroNascar {totalCarroNascar}
```



Ejemplo:

Pruebas	Salida
<pre>Carro carro[] = new Carro[5]; carro[0] = new Volqueta(100.0, 10.0); carro[1] = new Volqueta(200); carro[2] = new CarroNascar(150, 20.0); carro[3] = new CarroNascar(); carro[4] = new Volqueta(); PrecioTotal solucion = new PrecioTotal(carro); solucion.mostrarTotales();</pre>	<pre>Total Carro 19010.0 Total Volqueta 10920.0 Total Carro Nascar 8090.0</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 PrecioTotal {
    // Atributos

    // Constructores

    // Metodos
    public void calcularTotales() {
    }

    public void mostrarTotales() {
        // Calculo de totales
        calcularTotales();
        System.out.println("Total Carro " + totalPrecios);
        System.out.println("Total Volqueta" + totalVolqueta);
        System.out.println("Total Carro Nascar " + totalCarroNascar);
    }
}

public class Carro {
    // Constantes
    private static final double PESO = 10.0;
    private static final double TAMANIO = 4.5;
    private static final double PRECIO_BASE = 1000.0;

    // Atributos
    private double peso;
    private double tamano;
    private double precioBase;

    // Constructores

    // Metodos
    public double calcularPrecio(){
        return 0.0;
    }
    // getters/setters de ser necesarios
}
```



```
public class Volqueta extends Carro {  
    // Constantes  
    private static final double CAPACIDAD = 8.0;  
  
    // Constructores  
  
    // Metodos  
    public double calcularPrecio(){  
        // Calculos  
        return precioFinal;  
    }  
  
    // getters/setters de ser necesarios  
}  
  
public class CarroNascar extends Carro {  
    // Constantes  
    private final static int TIEMPO = 2;  
  
    // Constructores  
  
    // Metodos  
    public double calcularPrecio(){  
        // Calculos  
        return precioFinal;  
    }  
  
    // getters/setters de ser necesarios  
}  
// Fin de la solución
```

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