

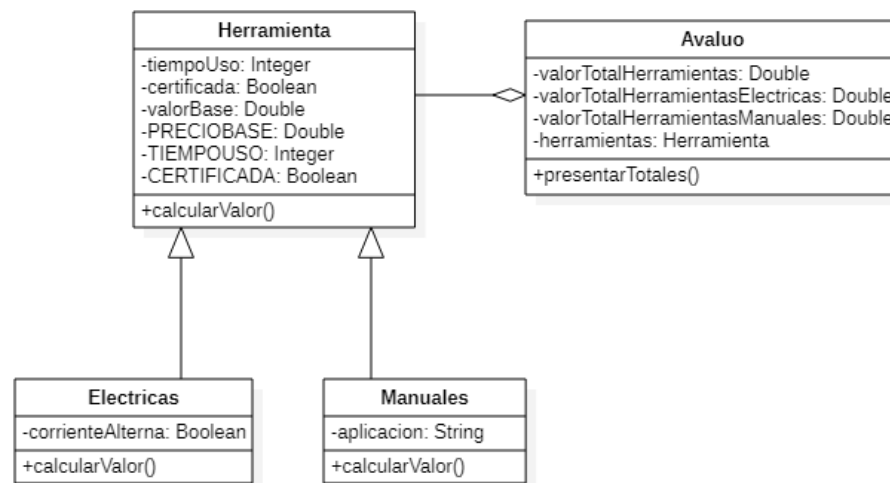


## Ciclo 2 Fundamentos de programación

### Reto 2

#### Descripción del problema:

Una empresa del sector de la manufactura ha decidido adquirir un conjunto de herramientas principalmente eléctricas, manuales y algunas genéricas. Con el fin de garantizar la inversión que va a realizar la empresa, necesita determinar el valor total del conjunto de herramientas, así como el valor total por tipo de herramientas. 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:



Todas las herramientas comparten los atributos **tiempoUso** y **certificada**, los cuales modifican su precio final. Se cuenta también con el atributo **valorBase** (en dólares), el cual representa el precio de la herramienta antes de sumar los respectivos valores según la evaluación de los atributos tiempo de uso y certificada. El método `calcularValor()` permite obtener el precio final para una herramienta de la siguiente forma:

Si la herramienta tiene entre 2 y 3 año (incluidos) de tiempo de uso al precio base de la herramienta disminuye un 10% de su precio base, si tiene mas de 3 años hasta 5 años (incluido) de tiempo de uso el precio base de la herramienta disminuye un 30% de su precio base, en caso de tener más de 5 años el precio base de la herramienta disminuye un 40% de su precio base, en caso de tener menos de 2 años no se disminuye su valor. Para el caso del atributo de certificada si este es verdadero la herramienta aumenta su precio base en USD\$30.

Las herramientas del tipo **Electricas** cuentan con el atributo adicional denominado **corrienteAlterna** el cual en caso de ser verdadero agrega un valor adicional al precio de la herramienta de USD\$20.

Las herramientas del tipo **Manuales** cuentan con el atributo adicional denominado **aplicacion** el cual en caso de ser TRAZO agrega un valor adicional al precio de la herramienta de USD\$10, en caso de ser MEDIDA agrega un valor adicional al precio de la herramienta de USD\$15 y si es CORTE agrega un valor adicional al precio de la herramienta de USD\$30.

En el conjunto existen herramientas que no son de tipo eléctricas ni de tipo manuales, estas serán tratados como herramientas de tipo **Herramienta** que son genéricas y su precio será calculado por el método `calcularValor()` de la clase **Herramienta**.



En caso de no ser suministrado se conoce que el mínimo precio base de cualquier herramienta es de USD\$25 y que el tiempo de uso mínimo supuesto de las herramientas es de 2 años y que no se encuentran certificadas. Estos valores deberán ser definidos como constantes de la clase `Herramienta` y se deben utilizar como valor por defecto para calcular el precio final de la herramienta en caso de que no se envíen estos valores al constructor de la clase. Por lo anterior debes implementar 2 constructores en la clase principal, uno en el que contemples los casos en que se envían todos los parámetros y otro en el que no se envía ningún parámetro.

Los atributos de la clase `ValorTotal` son: `valorTotalHerramientas`, `valorTotalHerramientasElectricas`, `valorTotalHerramientasManuales` y `herramientas`. Este último atributo contiene el conjunto de todas las herramientas, los cuales son almacenados en un array (tipo `Herramienta`) y son entregados al constructor de la clase `Avaluo` en el método `main()`, desde donde se llama al método `presentarTotales()`, el cual debe imprimir en consola:

El precio total de las herramientas que son genéricas es decir clase `Herramienta`.

El precio total de las herramientas tipo `Electricas`

El precio total de las herramientas tipo `Manuales`.

El precio total del conjunto de herramientas.

Notas:

- Los precios presentados deben ser redondeados con `Math.round()`.
- Cada una de las clases deben ser codificadas en una clase (archivo independiente) y las pruebas son ejecutadas desde en una clase `Main.java`.
- Las clases del reto, exceptuando `Main.java`, se deben cargar juntas en un solo archivo en la plataforma de pruebas `iMaster`.

### Ejemplo:

Pruebas	Salida
<pre>public class Main {     public static void main(String[] args) {         Herramienta[] herramientas = new Herramienta[5];         herramientas[0] = new Herramienta(3, true, 30.0);         herramientas[1] = new Electricas(3, true, 30.0, true);         herramientas[2] = new Manuales(3, true, 30.0, "TRAZO");         herramientas[3] = new Herramienta();         herramientas[4] = new Manuales(5, true, 90.0, "MEDIDA");         Avaluo respuesta = new Avaluo(herramientas);         respuesta.presentarTotales();     } }</pre>	80 77 175 332
<pre>public class Main {     public static void main(String[] args) {         Herramienta[] herramientas2 = new Herramienta[4];         herramientas2[0] = new Herramienta();         herramientas2[1] = new Herramienta(6, false, 120.0);         herramientas2[2] = new Electricas(7, false, 40.0, false);         herramientas2[3] = new Manuales(1, true, 60.0, "CORTE");         Avaluo respuesta2 = new Avaluo(herramientas2);     } }</pre>	95 24 120 239



```
        respuesta2.presentarTotales();  
  
    }  
}
```

### Esqueleto:

```
// Esta clase Main no debe ser subida a imaster solo es para que realice las  
pruebas  
  
public class Main {  
    public static void main(String[] args) {  
        Herramienta[] herramientas = new Herramienta[5];  
        herramientas[0] = new Herramienta(3, true, 30.0);  
        herramientas[1] = new Electricas(3, true, 30.0, true);  
        herramientas[2] = new Manuales(3, true, 30.0, "TRAZO");  
        herramientas[3] = new Herramienta();  
        herramientas[4] = new Manuales(5, true, 90.0, "MEDIDA");  
        Avaluo respuesta = new Avaluo(herramientas);  
        respuesta.presentarTotales();  
    }  
}  
  
class Herramienta {  
    // Constantes  
    private final static Double VALORBASE = 25.0;  
    private final static Integer TIEMPOUSO = 2;  
    private final static Boolean CERTIFICADA = false;  
  
    // Atributos  
    private Integer tiempoUso;  
    private Boolean certificada;  
    private Double valorBase;  
  
    // Constructores  
  
    // Metodos  
    public double calcularValor() {  
        //  
        return precioFinal;  
    }  
}
```



```
class Electricas extends Herramienta {
    // Atributos

    // Constructores

    // Metodos
    @Override
    public double calcularValor() {

        return precioFinal;
    }
}

class Manuales extends Herramienta {
    // Atributos

    // Constructores

    // Metddos
    @Override
    public double calcularValor() {
        return precioFinal;
    }
}

class Avaluo {
    // Atributos

    // Constructores

    public void presentarTotales() {
        // cálculos totales

        System.out.println(Math.round(valorTotalHerramientas));
        System.out.println(Math.round(valorTotalHerramientasElectricas));
        System.out.println(Math.round(valorTotalHerramientasManuales));
        System.out.println(Math.round(totalConjunto));
    }
}
```