

## Práctica 2

### Introducción al Diseño Orientado a Objetos

**Inicio:** Semana del 8 de febrero.

**Duración:** 2 semanas.

**Entrega:** Semana del 22 de febrero.

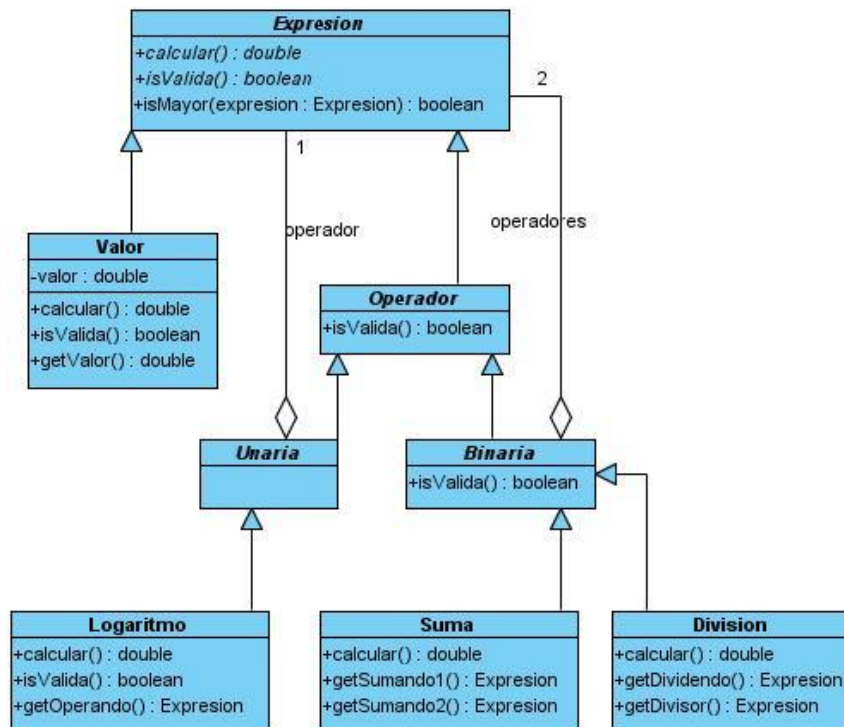
**Peso de la práctica:** 15%

El objetivo de esta práctica es introducir los conceptos básicos de la programación orientada a objetos desde el punto de vista del diseño, así como de su correspondencia en Java, con particular énfasis en el diseño de clases con relaciones de herencia, asociación y composición mediante diagramas de clases en UML.

### Apartado 1 (3 puntos):

Nos han encargado desarrollar en lenguaje Java las clases del siguiente diagrama de clases UML que se integrarán en una aplicación de cálculo matemático. La clase *Expresion* es una clase **abstracta** que representa todas las posibles operaciones matemáticas que se definirán en la aplicación. Una expresión puede ser un valor numérico (representado por la clase *Valor*) o un cálculo en el que intervienen uno (clase *Unaria*) o dos operadores (clase *Binaria*). Las expresiones con más de dos operadores se pueden definir recursivamente con objetos de estas tres clases. La funcionalidad que deben ofrecer las clases desarrolladas es:

- El cálculo de determinadas expresiones, en particular, valores numéricos, logaritmos, sumas y divisiones.
- Un comparador de operaciones que indique si una operación es mayor que otra en función de su valor.
- Un predicado que indique si la operación es o no válida.



### Se pide:

Desarrollar en Java las clases especificadas en el diagrama anterior de forma que tu implementación permita compilar y ejecutar el siguiente tester obteniendo la salida que se indica más abajo.

## Tester

```
package es.uam.eps.ads.p2.operaciones;

/**
 * Tester el primer apartado de la P2
 * @author Profesores ADS
 */
public class TesterOperaciones {

    public static void main(String[] args) {
        Expresion v1 = new Valor(3.44);
        Expresion v2 = new Valor(-6.33);
        Expresion v3 = new Valor(10.22);
        Expresion suma = new Suma(new Suma(v1, v2), v3);
        Expresion ln = new Logaritmo(new Division(suma, new Suma(v1, new Valor(-1.23))));
        Expresion error = new Logaritmo(new Suma(v1, v2));

        System.out.print("isValida " + suma + " ? " + suma.isValida());
        System.out.println(" ---> " + suma + " = " + suma.calcular());

        System.out.print("isValida " + ln + " ? " + ln.isValida());
        System.out.println(" ---> " + ln + " = " + ln.calcular());

        System.out.print("isValida " + error + " ? " + error.isValida());
        System.out.println(" ---> " + error + " = " + error.calcular());

        System.out.println();
        System.out.println(v1 + " mayor " + v3 + " ? " + v1.isMayor(v3));
        System.out.println(suma + " mayor " + ln + " ? " + suma.isMayor(ln));
    }
}
```

Siendo la salida esperada de la ejecución del programa la siguiente:

```
isValida 3.44 + -6.33 + 10.22 ? true ---> 3.44 + -6.33 + 10.22 = 7.33
isValida ln(3.44 + -6.33 + 10.22 / 3.44 + -1.23) ? true ---> ln(3.44 + -6.33 + 10.22 / 3.44 + -
1.23) = 1.1989830003688988
isValida ln(3.44 + -6.33) ? false ---> ln(3.44 + -6.33) = NaN

3.44 mayor 10.22 ? false
3.44 + -6.33 + 10.22 mayor ln(3.44 + -6.33 + 10.22 / 3.44 + -1.23) ? true
```

## Apartado 2 (3.5 puntos):

Tienes que diseñar el módulo de una aplicación de venta de electrodomésticos que se encarga de calcular el importe final que debe pagar el cliente en función del precio del producto, de si se debe realizar la entrega a domicilio y de si en el momento de la compra se entrega un electrodoméstico viejo. En la tienda se venden tres tipos de electrodomésticos de los que se deben almacenar los siguientes datos:

- *Televisión*: marca, modelo, precio base, tamaño de pantalla y clase energética (una letra entre A y D, ver más abajo).
- *Lavadora*: marca, modelo, precio base, dimensiones, clase energética y peso.
- *Frigorífico*: marca, modelo, precio base, dimensiones, clases energética y peso.

Si el cliente recoge el electrodoméstico en mano no se añade al precio ningún importe extra. Si se debe realizar la entrega a domicilio se aplican las siguientes reglas para calcular el coste de los portes. Este coste se debe añadir al importe a pagar por el cliente:

- Para las televisiones;
  - Si el tamaño de la pantalla es menor o igual a 40" y el precio de venta supera los 500€ la entrega será gratuita.
  - Si el tamaño de la pantalla es menor o igual a 40" y el precio de venta no supera los 500€ (sin tener en cuenta los posibles descuentos) el coste del porte serán 35€.
  - En cualquier otro caso, el coste del porte serán 35€ más 1€ por cada pulgada que la dimensión de la pantalla supere las 40". Por ejemplo, el coste de llevar al domicilio del cliente una televisión de 45" será 40€.

- Para las lavadoras:
  - Si el peso es inferior o igual a 50 kg el coste del porte serán 35€.
  - Si el peso supera los 50 kg el coste del porte serán 35€ más 0.50€ por cada fracción de 1 kg que el peso supere los 50 kg. Por ejemplo, el coste de llevar una lavadora de 56.5 kg al domicilio del cliente será 38.50€.
- El coste por entregar en el domicilio del cliente un frigorífico se calculará redondeando a valor entero el resultado de multiplicar su volumen (en m<sup>3</sup>) por 70€. Por ejemplo, el precio del porte de un frigorífico de 59.5 cm de ancho, 71.9 cm de fondo y 201 cm de alto será 60€.

Todos los electrodomésticos en stock en la tienda están catalogados con un valor de eficiencia energética en función de su consumo. La clase energética del electrodoméstico se representa con una letra entre la A y la D, siendo los electrodomésticos de clase A los de menor consumo (mayor clase energética) y los de clase D los que más consumen (menor clase energética). Si en el momento de la compra el cliente entrega un electrodoméstico viejo, se le aplicará un descuento en el importe a pagar en función de la clase energética del electrodoméstico que compra y del que entrega (independientemente del tipo de ambos electrodomésticos). Las reglas para calcular la cuantía de este descuento son:

- Si el electrodoméstico viejo es de la misma clase energética del que se adquiere, se aplicará un descuento de 25€ al precio final de la compra.
- Si es de una clase energética menor, a los 25€ de descuento se le añadirán 15€ por cada clase energética de diferencia respecto al nuevo electrodoméstico. Por ejemplo, si se compra una lavadora de clase energética B y se entrega una televisión de clase D el descuento a aplicar serán 55€.
- Si es de mayor clase energética, a los 25€ de descuento se le quitarán 5€ por cada clase energética de diferencia. Por ejemplo, si se compra una lavadora de clase energética D y se entrega una televisión de clase B el descuento será de 15€.
- Si se desconoce la clase energética del electrodoméstico viejo, se aplicará un descuento de 10€.

### **Se pide:**

Realizar un diagrama de clases UML para el nuevo módulo de la aplicación.

**Nota:** En el diagrama de clases en UML no es necesario incluir constructores, ni métodos *getters* y *setters*.

### **Apartado 3 (3.5 puntos):**

Añadir las modificaciones necesarias al diseño realizado en el ejercicio anterior para representar la tienda e incluir la siguiente funcional asociada a la gestión del stock:

- Añadir un nuevo modelo de electrodoméstico (televisión, lavadora o frigorífico)
- Aumentar el stock de un modelo de electrodoméstico
- Vender un electrodoméstico
- Descatalogar un electrodoméstico
- Mostrar el inventario de electrodomésticos, completo o por tipo

Sólo se deberá entregar el diagrama de clases UML modificado, no es necesario entregar la implementación *Java*.

### **Apartado 4 (Opcional, 1 punto):**

La tienda para la que has realizado el diseño de los apartados 2 y 3 ha decidido ampliar su negocio proporcionando servicio técnico y de solución de incidencias telefónico y un servicio de reparación de determinados electrodomésticos. Te han pedido que modifiques el diseño de la aplicación para dar soporte a estos nuevos servicios según los requisitos de negocio que se especifican a continuación.

El servicio técnico y de solución de incidencias sólo se ofertará para algunos modelos de televisión (no todos) que deberán identificarse a través de la aplicación. Al identificar los modelos de televisión para los que se va a ofrecer el servicio se indicará la url de la documentación técnica de la televisión para que el personal de soporte pueda acceder a ella y consultarla en caso de necesidad. Además de para televisiones, también se ofrecerá servicio técnico y solución de incidencias de tablets y teléfonos móviles, aunque se contempla la posibilidad de en un futuro también hacerlo para ordenadores sobremesa y portátiles. Para las tablets y los teléfonos móviles se deberá almacenar la siguiente información: marca, modelo y url de la documentación técnica. Este servicio sólo se proporcionará a aquellos clientes que lo contraten de forma específica pagando una tarifa anual que les dará derecho a realizar todas

las consultas que deseen durante el periodo de vigencia del contrato. El cobro de esta tarifa y la gestión de los clientes que tienen acceso al servicio la realizará la aplicación de facturación de la tienda.

Inicialmente se realizarán reparaciones de todo tipo de televisores y lavadoras. Todos los modelos de televisión y lavadora catalogados tendrán un precio específico por hora de reparación (es decir, que cada uno ellos podrá tener un precio diferente). La aplicación deberá calcular el coste de la reparación aplicando las siguientes reglas:

- Si el electrodoméstico se ha reparado satisfactoriamente, el coste se calculará multiplicando el número de horas empleado por el coste por hora de reparación del modelo de televisión o lavadora específico. Tanto el resultado de la reparación (satisfactoria/no satisfactoria), como el número de horas empleadas en ella se proporcionarán manualmente en el momento de la devolución del electrodoméstico al cliente.
- Si el electrodoméstico no ha podido repararse, no se cobrará nada al cliente.

### **Se pide:**

Incorporar al diseño de los ejercicios anterior los cambios necesarios para incluir la nueva funcionalidad.

### **Normas de Entrega:**

- Se deberán entregar los apartados 1, 2, 3 y, opcionalmente, el 4.
- La entrega la realizará uno de los alumnos de la pareja a través de Moodle.
- Si el ejercicio pide código Java, se entregará el código fuente, y la documentación generada con *JavaDoc*.
- Si el ejercicio pide un diagrama de diseño, se deberá entregar en PDF junto con una breve explicación (dos o tres párrafos a lo sumo).
- Se debe entregar un único fichero ZIP / RAR con todo lo solicitado, que deberá llamarse de la siguiente manera: GR<numero\_grupo>\_<nombre\_estudiantes>.zip. Por ejemplo Marisa y Pedro, del grupo 2261, entregarían el fichero: GR2261\_MarisaPedro.zip.
- La estructura de los ficheros entregados deberá estructurarse en directorios, uno por cada apartado.