

# ADSI

Programación Orienta a Objetos (P00) - Java



Instructor: Gustavo Adolfo Rodríguez Q.  
garodriguez335@misena.edu.co  
ADSI

# Actividad 5

En un archivo comprimado en zip o rar llamado actividad5\_xx.zip donde xx es su número de identificación envíe la solución de actividad en el link de la plataforma indicada por el instructor. Debe enviar el desarrollo de la actividad, el proyecto en Netbeans donde está su propuesta de solución.

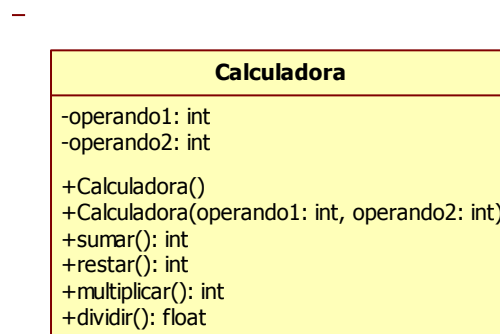
Desarrolle la actividad en clase y socialice activamente entre compañeros e instructor encargado.

Por favor tenga en cuenta las siguientes recomendaciones generales para el desarrollo de la actividad:

1. En el LMS del curso encontrará el Video Tutorial de la Actividad 5 , descargue este archivo en su disco duro local, descomprima el archivo y abra en el navegador web el archivo **calculadora\_demo.html**.

Siga el video tutorial y realice los pasos mostrados en el video tutorial de la misma forma como aparecen en la demostración.

Al finalizar el desarrollo de este video tutorial podrán notar que hemos implementado una clase llamada **Calculadora** como la mostrada en el siguiente diagrama de clases de UML.



2. Realice la implementación de una clase en Java llamada **MiCalculadora** que tenga los mismos métodos que la clase **Calculadora**, mostrada en el diagrama UML anterior, pero que además implemente los siguientes métodos.

Tenga en cuenta la descripción suministrada para cada método, recuerde la diferencia entre atributos de clase y argumentos de los métodos y recuerda utilizar la **herencia** como mecanismo para evitar la **redundancia** del código.

- `public int sumar() { ... } →` Éste método realizará la suma de los atributos `operando1 + operando2 + operando3` y retornará el resultado.
- `public int sumar(int sumando1, int sumando2) { ... } →` Éste método realizará la suma de los argumentos `sumando1 + sumando2` y retornará el resultado.
- `public int restar() { ... } →` Éste método realizará la resta de los atributos `operando1 - operando2` y retornará el resultado.
- `public int restar(int minuendo, int sustraendo) { ... } →` Éste método realizará la resta de los argumentos `minuendo - sustraendo` y retornará el resultado.
- `public int multiplicar() { ... } →` Éste método realizará la multiplicación de los atributos `operando1 * operando2 * operando3` y retornará el resultado.
- `public int multiplicar(int multiplicando, int multiplicador) { ... } →` Éste método realizará la multiplicación de los argumentos `multiplicando * multiplicador` y retornará el resultado.
- `public float dividir() { ... } →` Éste método realizará la división de los atributos `operando2 / operando3` y retornará el resultado.
- `public float dividir(int numerador, int denominador) { ... } →` Éste método realizará la división de los argumentos `numerador / denominador` y retornará el resultado.

Al finalizar la actividad deberá tener una implementación que corresponda con las clases del diagrama UML mostrada a continuación, y en el método `main` de la clase **Main** deberá tener un objeto de la clase **MiCalculadora** al que se le haya invocado todos sus métodos.

