TALLER #3

Utilizando la interface Operador, desarrollar una clase que evalúe una expresión aritmética. Dichas expresiones (y demás configuraciones) estarán contenidas en un archivo plano, enviando la ruta de éste por parámetro en la línea de comandos.

ENTRADA.

El archivo de entrada es un archivo en texto plano.
La primera línea del archivo contiene el número de expresiones N.
La segunda línea contiene M símbolos que pueden aparecer en las expresiones.
Las siguientes M líneas están separadas por espacios y pueden contener (en orden indiscriminado):

1. Cadena con el símbolo que va a representar.
2. Este segundo término puede contener dos tipos de información:
    a. El valor V equivalente al símbolo. Donde $-100 < V < 100$. Puede contener cifras decimales.
    b. Una cadena que contiene la clase que implementa la interface operador.
3. Precedencia del operador en caso de que el segundo campo sea una clase.

Después vienen las N expresiones a evaluar.

IMPLEMENTACIONES DE OPERADORES:

Para este ejercicio se debe implementar la interface Operador en las siguientes clases (La clase y el paquete deben llamarse igual):

1. **co.edu.poli.ces2.expresiones.taller2.SumaPoli:** Consiste en sumar a nivel de bits dos números (cada uno en valor absoluto) sin llevar el acarreo correspondiente. Por ejemplo, si quiere sumar 12 (1100 en binario) y 5 (101 en binario) nos da como resultado 9 (1001 en binario), ya que el tercer en el tercer bit no se tiene en cuenta el acarreo.

$$\begin{array}{r} 1100 \\ +0101 \\ \hline 1001 \end{array}$$

Ejemplos:

- 12 + 5 = 9
- 5.6 + (-4) = 1
- 7 + 9 = 14
- 50 + 10 = 56

Tener presente que:

- Si los valores tienen decimales, estos deben obviarse.
- No es una operación unaria.

2. **co.edu.poli.ces2.expresiones.taller2.Suma:**
    a. Operación Binaria: Suma aritmética

   b. Operación Unaria: Retorna el mismo valor.
3. **co.edu.poli.ces2.expresiones.taller2.Resta:**
   a. Operación Binaria: Resta aritmética
   b. Operación Unaria: Retorna el negativo del valor (es decir el valor * -1).
4. **co.edu.poli.ces2.expresiones.taller2.Multiplicacion:**
   a. Operación Binaria: Multiplicación aritmética
   b. Operación Unaria: No Aplica.
5. **co.edu.poli.ces2.expresiones.taller2.Division:**
   a. Operación Binaria: División aritmética
   b. Operación Unaria: No Aplica
6. **co.edu.poli.ces2.expresiones.taller2.Nmc:**
   a. Operación Binaria: No aplica
   b. Operación Unaria: Retorna el novecientos más cercano al valor dado. Si es menor que 900 retorna 900. El punto medio debe acercarse al valor mayor.
      Por ejemplo:
      i. Nmc(46863) = 46900
      ii. Nmc(-400) = 900
      iii. Nmc(1400) = 1900
7. **co.edu.poli.ces2.expresiones.taller2.Sqrt:**
   a. Operación Binaria: No Aplica.
   b. Operación Unaria: Calcula la Raíz cuadrada

SALIDA

En un archivo plano deben estar contenidos los N resultados (con 2 cifras decimales) de evaluar las expresiones. Si no se puede calcular un valor numérico debe aparecer NaN.

ENTREGABLES

- Archivo *.out que contiene el resultado de evaluar las expresiones. (1 resultado por línea). Redondeado a 2 decimales.
- Archivo .zip con los códigos fuentes del proyecto.

A TENER EN CUENTA

- Este taller vale por 2 notas en el seguimiento.
- Enviar al correo antes de las 10:00pm del viernes 7 de marzo.
- Es individual, no se engañe y procure desarrollar la actividad a conciencia.
- La idea es que el archivo *.jar se pueda ejecutar con el comando ***java –jar archivo.jar “ruta_archivo.in”*** y éste genere el archivo .out en la misma ruta.
- El archivo *.in que les envío es para que prueben su aplicación, para la evaluación se utilizará otro que cumpla con lo especificado en este documento.
- Es posible que el día viernes 7 les indique nuevas clases a implementar, para que las hagan en clase.
- El asunto del correo debe decir CES2 Taller#3 y el nombre del estudiante en el contenido.