

En un congreso de Aritmética decidieron utilizar un sistema para organizar sus expresiones matemáticas. Las expresiones con las que trabajan son expresiones básicas, que incluyen suma, resta, producto, división, potencia y raíz. Por ejemplo:

- $((2+3)*5)$
- $(5+5)^{(3-1)}$
- $((7+8)/(5-3))$



Luego de muchas discusiones se pusieron de acuerdo en qué es lo que querían que el sistema fuera capaz de proveer:

- Obtener el valor de la expresión: en los ejemplos anteriores, serían 25; 100; 7,5 respectivamente
- Obtener una lista con sólo los números de la expresión: en los ejemplos serían [2,3,5]; [5,5,3,1]; [7,8,5,3]
- Obtener una lista con sólo los operadores de la expresión: en los ejemplos serían [+,*]; [+pot,-]; [+/, -]
- Imprimir fácilmente las expresiones, por ello quieren que poder obtener un String con la expresión propiamente dicha incluyendo los paréntesis " $((2+3)*5)$ "; " $(5+5)(3-1)$ "; " $((7+8)/(5-3))$ ".

Adicionalmente, el sistema debe ser capaz de retornar el cálculo opuesto de una expresión. Esto quiere decir, si la expresión es una suma, el cálculo opuesto hace la resta, si es un producto hace la división, si es una potencia hace la raíz cuadrada, y así con cada operador. Por ejemplo, en la primera expresión presentada el cálculo opuesto sería $((2-3)/5)$ es decir -0,2. En el segundo ejemplo, el cálculo opuesto sería $^{3+1}\sqrt{(5-5)}$ es decir 0. En el tercer ejemplo, el cálculo opuesto sería $((7-8) * (5+3))$, es decir -8.

A uno de los matemáticos de la reunión le molestaba la posibilidad de que todo fallara al calcular ciertas expresiones, si la misma generaba un error matemático, tales como cuando intentaban una división por cero o calcular la raíz de un número negativo. Se resolvió entonces que, cuando el cálculo de una expresión genera un error matemático, se debe devolver el número "-23". (Si bien este número puede variar en un futuro, debe ser el mismo para todas las expresiones).

Los matemáticos desean realizar un mecanismo en el cual sea posible "normalizar" los valores de una expresión, es decir si tenemos la expresión $(4+5)$ y se la normaliza con el valor 10 la expresión resultante sería $(0,4 + 0,5)$. Si se la normaliza con el valor 2, la expresión resultante sería $(2 + 2,5)$. Para realizar esta operación, se genera una expresión nueva y la original no se modifica. La expresión generada debe respetar la misma estructura de la expresión original.

Finalmente, se debe incorporar la capacidad de generar una copia de una expresión. Cuando se duplica una expresión se realiza una copia de todos los elementos contenidos en ella.

- Implementar la solución al problema del organizador de expresiones en JAVA, tener en cuenta todos los conceptos vistos en clase. Prestar especial atención a la abstracción de funcionalidad.
- Implementar en un main la construcción de los tres ejemplos anteriores.