[ANALIZADOR]



1. Descripción de la notación post fija
2. Descripción de la clase StringTokenizer
3. Descripción de los métodos a usar

## 1. Notación post fija

Es un método algebraico alternativo de introducción de datos. Su nombre viene del matemático polaco Jan Lukasiewicz que en 1920 introdujo una notación análoga.

En la notación polaca primero están los operandos y después está el operador con el que se va a realizar el cálculo.

Si se utiliza una estructura LIFO la evaluación de una expresión se optimiza. (Ventaja, analizar otras)

En los sistemas educativos la notación algebraica es casi universal & esto hace que no se utilice la anterior (Desventaja, analizar otras)

Ejemplo: $5 + ((1+2) * 4) - 3$

$5\ 1\ 2 + 4\ * + 3\ -$

1° 2° 3°

Secuela: $5 + 7$

$5\ 7 +$

③ Mètodes a usar.

Mirar el diagrama UML. Comentar de abajo a arriba.

Evaluador: s una clase con una propiedad pila.
En el constructor la instanciamos, creando una pila vacia Integer

Mètodo privado esOperador: analiza si un string es un operador.
equals

Mètodo privado soloDigitos: analiza si un string se compone de solo digitos.
Character.isDigit(cadena)

Mètodo ~~evaluar~~ procesarUnOperador: mirar test

El operador que se pasa como parámetro se aplica a dos operandos que se sacan de la pila, el resultado se vuelve a meter en la pila

Mètodo eliminarBlancos

Usa un objeto de la clase stringTokenizer y un objeto de la clase cola.

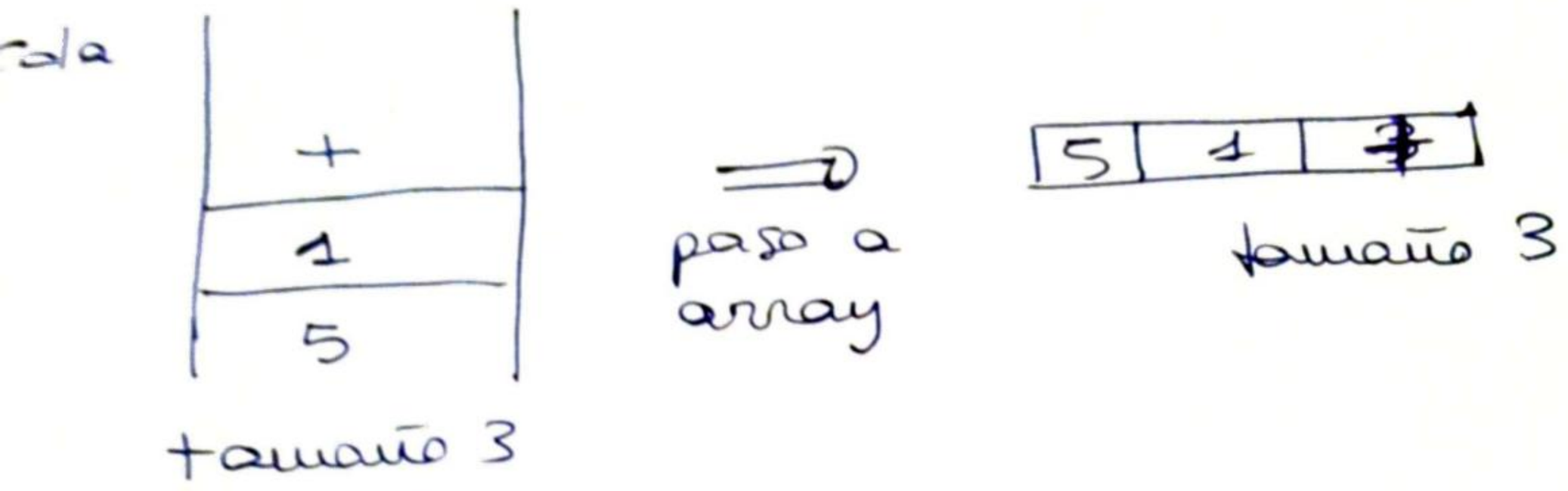
- El de la clase stringTokenizer se instancia con la expresión a evaluar, los delimitadores de la operaciones y el carácter blanco, la bandera se pone a verdadero.
- El objeto cola se instancia a cola vacia.

↓ ~~Integer~~ String

> Se recorre el objeto stringTokenizer hasta que no haya más elementos.
En cada iteración si el token no es blanco se mete en la cola.

> Se crea un array con el mismo nº de componentes que la cola anteriormente creada.
Guarda en el array los elementos que va sacando de la cola que ya no tienen blancos

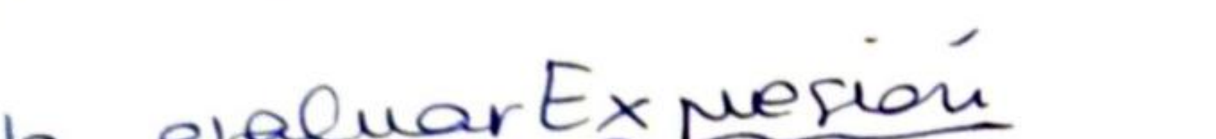
Ejemplo 5 b b 1 + b




Método evaluarExpresión

> Elimina b blancos de una expresión y lo guarda en un array. Usa el método anterior

5 b b 1 2 + b 4 * + 3 b -




> Vuelve a pasar el array a string para ver si solo son dígitos. En ese caso lo mete en la pila sin más.

> Si no son solo dígitos:

Recorre el array completo:

- Si el token es un operador, se pasa ⇒ método procesarUnOperador
- Si no mete el operando en la pila.

Cuando se acaba el recorrido, se devuelve el elemento de la pila.

Ejemplo: Array

| 5 | 4 | 2 | + | 4 | * | + | 3 | - |
|---|---|---|---|---|---|---|---|---|

String "5 4 2 + 4 * + 3 -"

String 4 "7845" ⇒ meto pila