

ESTRUCTURAS DE DATOS

Curso 2020 - 2021

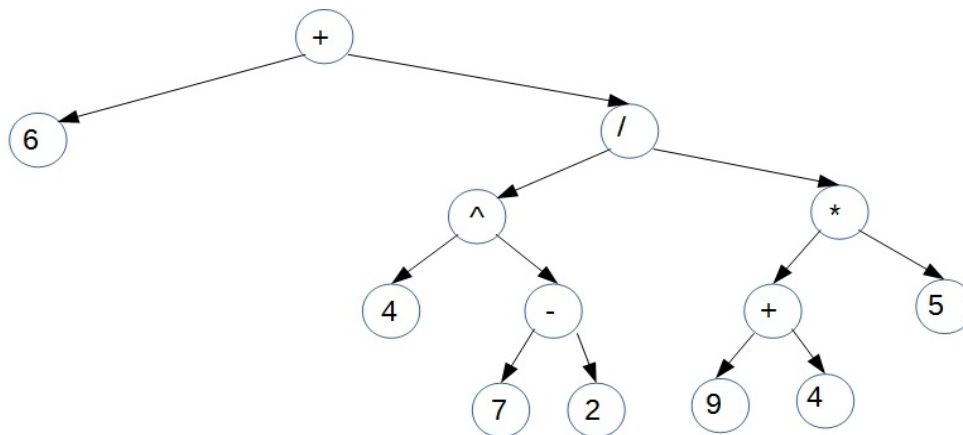
PRÁCTICA 6

Árboles binarios: expresiones aritméticas binarias

Instrucciones

- Se debe completar en una sesión.
- Práctica individual.
- Lee el enunciado completo antes de comenzar. Los comentarios incluidos en el código también pueden proporcionar información útil.
- Se habilitará una tarea para que entregues el código desarrollado.
- La práctica será APTA si se superan todos los test de validación proporcionados.

Se van a usar árboles binarios para representar expresiones aritméticas binarias. Para simplificar, los operadores utilizados en las expresiones son $+$, $-$, $/$, $*$ y $^$ (potencia) y los números serán sólo dígitos del 0 al 9. Por tanto, los árboles binarios podrán representar expresiones del tipo: $3+4$, $(6-2)*5$, $(8+4)*(9-3)$, $6+(4^(7-2)/(9+4)*5)$, etc. El árbol binario que representa esta última expresión sería:



Se os proporciona un fichero **EDBinaryNode.java** con una implementación genérica del nodo de árbol binario, Para representar la expresión se utiliza un árbol binario de caracteres. El fichero **BinaryExpressionTree.java** contiene la implementación de los métodos para trabajar con árboles expresiones. En esta práctica sumiremos que las expresiones se presentarán como cadenas de código y que todas están formadas correctamente. Una cadena formada por un sólo dígito es una expresión binaria correcta (por ejemplo, '5'). La clase contiene métodos para convertirlas en árboles binarios y calcular valor.

Ejercicio 1

Implementa un método estático de la clase **BinaryExpressionTree** que devuelva un **boolean** indicado si un árbol binario es extendido o no.

```
static boolean isExtended(EDBinaryNode<Character>)
```

Ejercicio 2

Implementa un método estático de la clase **BinaryExpressionTree** que devuelva **true** si un árbol binario de caracteres tiene dígitos entre '0' y '9' sólo en sus nodos hoja y los nodos internos contienen operadores válidos.

```
static boolean digistsOnLeaves(EDBinaryNode<Character> tree)
```

Ejercicio 3

Implementa en la clase **BinaryExpressionTree** un método estático que, a partir de un árbol binario con una expresión aritmética, devuelve el resultado de evaluar dicha expresión. Para el árbol de la figura, el método devolverá 26,48.

```
static float evaluate(EDBinaryNode<Character> tree)
```

Ejercicio 4

Implementa el método estático de la clase **BinaryExpressionTree** que devuelve una lista con la expresión aritmética que contiene el árbol siguiendo una notación infija (inorden) completamente parentizada. Por ejemplo, en el caso del árbol de la figura devolverá una lista con los siguientes nodos: (, 6, +, (, (, 4, ^, (, 7, -, 2,),),), /, (, (, 9, +, 4,), *, 5,),),), que corresponde a la expresión: $(6 + ((4^7 - 2) / ((9 + 4) * 5)))$.

```
static List<Character> asListInorder(EDBinaryNode<Character> tree)
```

Nota: Para las pruebas del test de esta práctica te proporcionamos el fichero de texto *expresiones.txt*. Este fichero debe ser almacenado en la carpeta raíz del proyecto para que puede ser encontrado

