

Algoritmos y Estructuras de Datos

Enunciado Primer Parcial Año 2015

Realizar un programa que evalúe una expresión aritmética con notación posfija. La expresión contiene solamente números enteros y las cuatro operaciones aritméticas básicas. La expresión está compuesta por una serie de “*tokens*”, donde cada *token* es un número o un operador aritmético.

La calculadora usa una lista donde se almacenan los *tokens* de la expresión y una pila para realizar el cómputo. La especificación (incompleta) de la calculadora es la siguiente:

Tipo Calculadora

Usa Lista, Pila, Entero, Booleano

Operaciones:

<i>nvaCalc</i> :	<i>Lista</i>	\longrightarrow	<i>Calculadora</i>
<i>verL</i> :	<i>Calculadora</i>	\longrightarrow	<i>Lista</i>
<i>verP</i> :	<i>Calculadora</i>	\longrightarrow	<i>Pila</i>
<i>evaluar</i> :	<i>Calculadora</i>	\longrightarrow	<i>Entero</i>
<i>avanza</i> :	<i>Calculadora</i>	\longrightarrow	<i>Calculadora</i>
<i>opera</i> :	<i>Calculadora</i>	\longrightarrow	<i>Calculadora</i>
<i>calcula</i> :	<i>Operad</i> \times <i>Entero</i> \times <i>Entero</i>	\longrightarrow	<i>Token</i>
<i>esNum</i> :	<i>Token</i>	\longrightarrow	<i>Booleano</i>
<i>esOpe</i> :	<i>Token</i>	\longrightarrow	<i>Booleano</i>
<i>verificar</i> :	<i>Calculadora</i>	\longrightarrow	<i>Booleano</i>

Comportamiento:

l, l' :Lista p, p' :Pila c :Calculadora

$$\begin{aligned}
\text{verL}(\text{nvaCalc}(l)) &= l \\
\text{verP}(\text{nvaCalc}(l)) &= \text{nuevaPila}() \\
\text{evaluar}(c) &= \begin{cases} \text{ERROR} & \text{verificar}(c) = \text{false} \\ \text{tope}(\text{verP}(c)) & \text{esvacia}(\text{verL}(c)) = \text{true} \\ \text{evaluar}(\text{avanza}(c)) & \text{esNum}(\text{cabeza}(\text{verL}(c))) = \text{true} \\ \text{evaluar}(\text{opera}(c)) & \text{esOpe}(\text{cabeza}(\text{verL}(c))) = \text{true} \end{cases} \\
\text{verL}(\text{avanza}(c)) &= \text{resto}(\text{verL}(c)) \\
\text{verP}(\text{avanza}(c)) &= \text{apilar}(\text{verP}(c), \text{cabeza}(\text{verL}(c))) \\
\text{verL}(\text{opera}(c)) &= \text{resto}(\text{verL}(c)) \\
\text{verP}(\text{opera}(c)) &= \text{apilar}(\text{desapilar}(\text{desapilar}(\text{verP}(c))), \\
&\quad \text{calcula}(\text{cabeza}(\text{verL}(c)), \text{tope}(\text{verP}(c)), \text{tope}(\text{desapilar}(\text{verP}(c))))
\end{aligned}$$

Deberá codificar la solución usando Clases de C++, con métodos que respeten el comportamiento definido en la especificación, tanto en parámetros como en la operatividad.

La función verificar no está definida y queda a su libertad la implementación. Deberá devolver falso si la expresión posfija no respeta la relación entre el número de valores y el de operadores. La función calcular es la que realiza el cálculo de la operación aritmética.

La clase Calculadora tiene como parámetro en su constructor a la lista que contiene la expresión posfija. El contenido de dicha lista debe ser definido en el método *main* del programa.

Podrá utilizar el código de las clases lista y pila dados en clases prácticas. El tipo de dato contenido en la clase lista y pila debe ser el mismo. Es conveniente que cree una clase *Token* que almacene el operador y el valor y que tenga un método que determine si el token es de valor o de operador. La clase Lista y la Clase Pila tendrían en este caso a *Token* como el tipo de dato contenido.