

Universidad Nacional de Córdoba

Facultad de Ciencias Exactas, Físicas y Naturales

**Algoritmos y Estructuras de Datos**

Parcial N°1

Calculadora Postfija

Alumno:

Heredia, Marco

Matricula:

93560445

Fecha:

14 de mayo de 2015

La consigna pedía realizar un programa que evalué una expresión aritmética con notación posfija, esta expresión podía contener números enteros y las cuatro operaciones aritméticas elementales.

Para realizarlo, utilice las clases listas y pilas vistas en clase y en ciertos métodos los modifique convenientemente para la realización de la calculadora.

El programa consta de cinco clase: Calculadora, Pila, Lista, Nodo y Token las cuales son enlazadas una vez que se las llama en la función main.

*Breve descripción de cada clase implementada*

Las clases Pila, Lista y Nodo fueron utilizadas tal cual las vi en clase por lo que no las voy a describir, a excepción de 2 métodos de la clase lista que tuve que agregar los cuales voy a describir mas adelante.

Clase Token:

La clase token es el tipo de dato que va a estar contenido en las pilas y listas. El constructor de la clase se inicializa con una cadena de texto String y la misma evalúa el primer carácter del String, en caso de ser uno de los operadores básicos (+ - \* /) guarda ese operador como un char y su variable se identifica como un operador, en el caso que el primero carácter no sea un operador, el String pasado va a tener que ser un numero y el mismo lo guarda como double, decidí guardarlo como doble y no como entero como dice la consigna del parcial para permitir un resultado exacto entre la división de enteros. En esta clase y en todo el programa no se contempla el hecho que se pase algún otro carácter distinto de un número o de las 4 operaciones elementales.

La clase tiene 3 métodos: uno para saber si el Token en cuestión es un número o un operador, uno para obtener el operador y otro para obtener el número, dependiendo que tipo de dato contiene.

Clase Lista:

Utilice la clase lista vista en las clases prácticas de la materia y le agregue 2 métodos: cantOpe() y cantNum() ambos métodos están diseñados para funcionar en conjunto con la clase Token, los métodos recorren la lista desde el principio hasta el final y van aumentando su contador siempre que encuentran un operador o un numero respectivamente. Para ello utilizan el método es\_ope() de la clase Token.

Clase Calculadora:

La clase calculadora utiliza punteros a una lista y a una pila. La lista la obtiene en el constructor de la clase, en esa lista esta contenido una expresión matemática postfija, la pila al principio esta vacía y en esta pila se van a ir insertando los números y sobre la pila se va a ir operando.

La calculadora tiene métodos que devuelven un puntero a la lista y a la pila llamados verL() y verP() respectivamente.

Tiene método llamada verificar() que devuelve un booleano true en caso que la cantidad de números sea uno superior que la cantidad de operadores (ya que de esta forma nos aseguramos una cantidad de operadores justa para la cantidad de números) y devuelve false en caso que esto no se cumpla. Para facilitar la verificación agregue en la clase lista 2 funciones que se encargar de realizar el conteo de la cantidad de números y operadores que contiene dicha lista.

El método evaluar se encargar de realizar el calculo de la expresión postfija ingresado en la lista. Este método verifica que la expresión postfija sea correcta y en caso de que sea correcta empieza a evaluar token por token desde el comienzo hasta el final. Si el token actual es un numero, lo inserta en la pila y salta al siguiente token de la lista con el método avanza(). Si el token actual es un operador realiza la operación de los últimos 2 números ingresados en la pila, desapila los 2 números y apila un token con el resultado de la operación con el método opera().

El método repite estos pasos hasta que la lista este vacía, en ese momento devuelve el valor que se encuentra en la pila, el cual es el resultado final de la expresión postfija.

Función main:

En la función main se inicializa una lista y se le agrega la expresión postfija con el método add() de la clase lista. Debido a que este método agrega nodos al principio de la lista, la expresión postfija se debe ingresar al revés.

Una vez que termina de ingresar la expresión a la lista, muestra en pantalla la lista completa.

Luego inicializa la calculadora y le pasa como parámetro la lista previamente cargada con los datos de la expresión matemática y llama al método evaluar() de la calculadora para realizar el calculo de la expresión.

Por último muestra en pantalla el valor que quedo en la pila de la calculadora, el cual es el resultado final.