

Actividades Estructura de Datos Pila

Actividad 1

Se desea desarrollar una aplicación para la gestión de una pila de cadenas de caracteres. La interfaz de usuario consistirá en un menú que aparecerá en pantalla al arrancar el programa, y cuyas opciones serán las siguientes:

1. Agregar cadena
2. Quitar cadena
3. Mostrar todas
4. Salir

La opción 1 añade una nueva cadena a la pila (el tamaño de la pila es ilimitado) teniendo en cuenta que las cadenas no se pueden repetir, por otro lado, la opción 2 elimina una cadena de una pila (la última en llegar), mientras que la opción 3 muestra en pantalla todas las cadenas. Tras realizar la operación correspondiente, el programa vuelve a mostrar el mismo menú inicial, algo que se repetirá hasta que el usuario elija la opción salir.

Para acometer el desarrollo de esta aplicación se sugiere la filosofía planteada anteriormente, consistente en la separación entre «la lógica de aplicación» y las operaciones de entrada y salida. Para ello se sugiere la creación de una clase Pila donde se encapsularán todas las operaciones sobre la pila. Aunque la aplicación se centra en el tratamiento de pilas de caracteres, dicha clase Pila deberá implementarse como tipo genérico, de modo que pueda permitir el apilamiento de cualquier tipo de objetos.

Métodos que deberá tener la clase Pila son:

void push(E cad)

E pop()

void isEmpty()

E obtener(int posicion). Devuelve el elemento que ocupa la posición indicada, considerando como 0 la posición del primer objeto de la pila.

En otra clase, llamada Principal, se incluirá el método main() que incluirá todas las instrucciones necesarias para la generación del menú y del resto de las operaciones de entrada y salida de datos que deba incluir la aplicación.

Actividad 2

La profundidad de anidamientos en un punto particular de una expresión es la cantidad de ámbitos que se han abierto pero que todavía no se han cerrado en este punto. Es lo mismo que la cantidad de paréntesis izquierdos encontrados cuyos correspondientes paréntesis derechos todavía no han aparecido. Definimos el conteo de paréntesis en un punto particular de una expresión como la cantidad de paréntesis izquierdos, menos la cantidad de paréntesis derechos que se han encontrado, examinando la expresión a partir de su extremo izquierdo hasta un punto particular. Si el conteo de paréntesis es no negativa, es igual que la profundidad de anidamiento. Las siguientes son las dos condiciones que deben prevalecer si los paréntesis de una expresión forman un patrón admisible:

1. El conteo de paréntesis al final de la expresión es 0. Esto implica que no han quedado ámbitos abiertos o que se ha encontrado exactamente la misma cantidad de paréntesis derechos e izquierdos.
2. El conteo de paréntesis en cada punto de la expresión es no negativa. Esto implica que no se ha encontrado un paréntesis derecho para el cual no se haya encontrado el correspondiente paréntesis izquierdo.

Cadena de ejemplo:

Expresión	7	-	((X	*	((X	+	Y)	/	(J	-	3))	+	Y)	/	(4	-	2.5))
Conteo de paréntesis	0	0	1	2	2	2	3	4	4	4	4	3	3	4	4	4	4	3	2	2	2	1	1	2	2	2	2	1	0

Ahora supóngase que existen tres tipos de delimitadores de ámbito. Estos tipos se identifican mediante paréntesis ((y)), paréntesis cuadrados ([y]) y llaves ({ y }). Un símbolo para cerrar un ámbito debe ser del mismo tipo usado para abrirlo. Por tanto cadenas como

$$(A+B), [(A+B)], \{A-(B)\}$$

no son válidas.

Es necesario registrar no sólo cuántos ámbitos se han abierto sino también sus tipos. Esta información se necesita porque cuando se encuentra un símbolo para cerrar un ámbito, debemos conocer el símbolo con el que se abrió para comprobar que se cierra en forma adecuada.

Debe usarse una pila para registrar los tipos de ámbito encontrados. Cuando se encuentra un símbolo para abrir un ámbito, se agrega a la pila. Cuando se encuentra un símbolo para cerrar un ámbito, se examina la pila. Si está vacía, el símbolo no tiene el signo correspondiente para abrir, y por tanto, la cadena no es válida. Sin embargo, si la pila no está vacía, removemos el elemento de la pila y se verifica si el elemento removido corresponde al símbolo para cerrar el ámbito. Si coinciden, se prosigue. En caso contrario, la cadena no es válida. Cuando se llega al fin de la cadena, la pila debe estar vacía; de otra forma, se han abierto uno o más ámbitos que no se han cerrado y la cadena no es válida. El último ámbito en abrirse debe ser el primero en cerrarse. Elabore un programa que, usando la definición de pila vista en clase, valide las cadenas tomando en consideración lo expuesto anteriormente.