

Tipo de Datos Abstracto *PILA*, EJERCICIO TA6

Ejercicio 1, requerimiento:

Se necesita construir un analizador sintáctico para cierto lenguaje de programación y para ello es necesario escribir un método que, dada una entrada representada por la lista de caracteres del código fuente, controle si la secuencia de corchetes es correcta o no.

Por ejemplo,

Es una secuencia **bien formada**: **{{{}}**

Es una secuencia **mal formada**: **{{{}}**

Con la ayuda del tipo de datos abstracto **PILA**, escribe en pseudocódigo la siguiente funcionalidad:

De tipo **booleano controlCorchetes**(de tipo lista de caracteres *listaDeEntrada*)

Lenguaje Natural:

Partir de una pila vacía. Recorrer la lista de caracteres: cuando se encuentra un carácter “{” realizar una inserción en la pila (**Mete, Push**) de un elemento (verificando que la pila no esté llena, de lo contrario no se puede terminar el proceso por falta de espacio suficiente). Cuando se encuentra un carácter “}”, procurar realizar una extracción de la pila (**Saca, Pop**). Si en este momento se produce un error (por pila ya vacía) entonces la expresión no está bien formada (se trata de cerrar más veces que las que se ha abierto correspondientemente). Si se llega al final de la lista de caracteres sin errores, la pila debe quedar vacía (caso expresión bien formada) o de lo contrario la expresión está mal formada (faltan cierres).

PRECONDICIONES.

- La pila debe estar vacía (esto se aplica en realidad si vamos a usar una pila pre-existente)
- Opcional: la lista de caracteres no está vacía (esto es discutible: ¿la expresión vacía es válida?)

POSTCONDICIONES:

- La lista de caracteres permanece inalterada
- Si la expresión está bien formada, la pila vuelve a estar vacía.

SEUDOCÓDIGO

tipo booleano **controlCorchetes**(de tipo lista de caracteres **listaDeEntrada**)

COM

tipoPila laPila // si usamos una pila preexistente debe estar vacía

Si listaDeEntrada.vacia

 Devolver error "expresión vacía" // devolver termina

AuxLista ← listaDeEntrada.primerO

Si AuxLista.Siguiente = nulo **devolver FALSO** // 1 sólo carácter está mal formada

Mientras AuxLista.Siguiente <> nulo

Si AuxLista.Etiqueta = "{"

Si laPila.Mete(unElementoCualquiera)= FALSO **devolver FALSO** //pila llena

SiNo

Si AuxLista.Etiqueta = "}"

Si laPila.Saca = nulo **devolver FALSO** // pila vacía antes de "sacar"

 AuxLista ← AuxLista.Siguiente

FinMientras

//llegamos al final de la lista de caracteres

Si laPila.vacia **devolver VERDADERO**

SiNo **devolver FALSO**

FIN

Operaciones del TDA Pila utilizadas:

- TipoElemento **Saca** // devuelve el elemento que está en el tope de la pila, o nulo si la pila está vacía
- Tipo booleano **Mete**(TipoElemento unElemento) // devuelve Verdadero si se pudo colocar el elemento en la pila, o FALSO de lo contrario (por ejemplo, pila llena)
- Tipo booleano **Vacia**