



UNIVERSIDADE DA CORUÑA

Programación II

TAD Pila

Especificación informal TAD Pila

TAD Pila

VALORES:

- Una pila es una secuencia de cero o más elementos de un mismo tipo $(a_1, a_2, a_3, \dots, a_n)$ donde $n \geq 0$:
 - Si $n=0$ se dice que la pila es vacía.
 - Los elementos de la pila están ordenados de forma lineal, no por su contenido, sino por la posición que ocupan. Es decir:
 - a_i precede a a_{i+1} ($\forall i, i=1 \dots n-1$)
 - a_i sucede a a_{i-1} ($\forall i, i=2 \dots n$)

OPERACIONES (SINTAXIS y SEMÁNTICA)

- Generadoras
 - $\text{PilaVacía} \rightarrow \text{Pila}$
{*Objetivo*: Crea una pila vacía
Salida: Una pila vacía
Poscondicion: La pila sin datos}
 - $\text{Apilar}(\text{Dato}, \text{Pila}) \rightarrow \text{Pila}, \text{Boolean}$
{*Objetivo*: Mete un elemento en la pila quedando en la cima
Entrada:
Dato: Contenido del elemento a meter
Pila: Pila donde vamos a meter
Salida:
Pila: Pila con el elemento Dato en la cima y verdadero si se ha podido meter, falso en caso contrario}

■ Destructoras

● DesApilar (Pila) \rightarrow Pila

{*Objetivo*: Saca el elemento de la cima de la pila

Entrada:

Pila: Pila de donde vamos a sacar

Salida:

Pila: Pila sin el elemento de su cima

Precondición: La pila no está vacía}

■ Observadoras

● CimaPila (Pila) \rightarrow Dato

{*Objetivo*: Recupera el contenido del elemento de la cima de la pila

Entrada:

Pila: Pila donde obtener el dato

Salida:

Dato: Contenido del elemento de la cima de la pila

Precondición: La pila no está vacía}

● EsPilaVacia (Pila) \rightarrow Boolean

{*Objetivo*: Determina si una pila está vacía

Entrada:

Pila: Pila a comprobar

Salida:

Verdadero si la pila está vacía, falso en caso contrario}

Aplicaciones de las pilas: Equilibrio de símbolos

- Una pila es una herramienta útil para comprobar el equilibrio de símbolos como $()$, $\{\}$, and $[]$.
- Por ejemplo, es legal la secuencia

$$\dots [\dots (\dots) \dots] \dots$$

pero no así la secuencia

$$\dots [\dots (\dots) \dots] \dots$$

- La presencia de un símbolo mal colocado podría ocasionar cientos de mensajes de error del compilador carentes de sentido alguno.

Seudocódigo

1. Crear una **pila vacía**.
2. Leer tokens hasta el fin del archivo (EOF).
 - Si el token leído es de apertura se mete en la pila
 - Si el token leído es de clausura
 - Si la pila está vacía se genera un error
 - Si la pila no está vacía, se saca un elemento de la pila
 - Si el token de apertura no es el correspondiente al de clausura se genera un error
 - Si no continuar
3. Si la pila no está vacía, se genera un error

Ejemplo: Aplicar este algoritmo a la sentencia:

$$s = t[5] + u / (v * (w + y));$$