

Pilas

Una pila es un tipo especial de lista en la que todas las inserciones y supresiones tienen lugar en un extremo denominado tope. A las pilas se les llama también «listas LIFO» (last in first out) o listas «último en entrar, primero en salir». El modelo intuitivo de una pila es precisamente una pila de fichas de póquer puesta sobre una mesa, o de libros sobre el piso, o de platos en una estantería, situaciones todas en las que sólo es conveniente quitar o agregar un objeto del extremo superior de la pila, al que denominará en lo sucesivo «tope». Un tipo de datos abstracto de la familia PILA incluye a menudo las cinco operaciones siguientes:

1.- ANULA (P) convierte la pila P en una pila vacía. Esta operación es exactamente la misma que para las listas generales.

2.- TOPE (P) devuelve el valor del elemento de la parte superior de la pila P. Si se identifica la parte superior de una pila con la posición 1, como suele hacerse, entonces TOPE (P) puede escribirse en función de operaciones con listas como RECUPERA(PRIMERO(P),P).

3.- SACA (P), en inglés POP, suprime el elemento superior de la pila, es decir, equivale a SUPRIME (PRIMERO(P),P). Algunas veces resulta conveniente implantar SACA como una función que devuelve el elemento que acaba de suprimir, aunque aquí no se hará eso.

4.- METE (x, P), en inglés PUSH, inserta el elemento x en la parte superior de la pila P. El anterior tope se convierte en el siguiente elemento, y así sucesivamente. En función de operaciones primitivas con listas, esta operación es INSERTA (x, PRIMERO(P),P).

5.- VACIA (P) devuelve verdadero si la pila P está vacía, y falso en caso contrario.

Especificaciones de una pila

Las operaciones que sirven para definir una pila y poder manipular su contenido son las siguientes.

Tipo de dato	Dato que se almacena en la pila
Operaciones	
CrearPila	Inicia la pila.
Insertar (Push)	Pone un dato en la pila.
Quitar (Pop)	Retira (saca) un dato de la pila.
Pilavacía	Comprobar si la pila no tiene elementos.
Pilallena	Comprobar si la pila está llena de elementos.
Limpiar pila	Quita todos sus elementos y dejar la pila vacía.
CimaPila	Obtiene el elemento cima de la pila.
Tamaño de la pila	Número de elementos máximo que puede contener la pila.

La operación Pilallena sólo se implementa cuando se utiliza un array para almacenar los elementos. Una pila puede crecer indefinidamente si se implementa con una estructura dinámica.

Representación de pilas

Las pilas no son estructuras fundamentales de datos; es decir, no están definidas como tales en los lenguajes de programación. Para su representación requiere el uso de otras estructuras de datos, como:

- Arreglos

- Listas

En consecuencia, es importante definir el tamaño máximo de la pila, como la variable auxiliar a la que se denomina TOPE. Esta variable se utiliza para indicar el último elemento que se insertó en pila.

Realización de pilas basadas en arreglos

Todas las realizaciones de listas descritas sirven para pilas puesto que una pila con sus operaciones es un caso especial de lista con sus operaciones. La representación de una pila como lista enlazada es sencilla, pues METE y SACA operan sólo con la celda de encabezamiento y con la primera celda de lista.

Un mejor criterio para usar un arreglo es el que tiene en cuenta el hecho de que las inserciones y las supresiones ocurren sólo en la parte superior. Se puede anclar la base de la pila a la base del arreglo (el extremo de índice más alta) y dejar que la pila crezca hacia la parte superior del arreglo (el extremo de índice más bajo). Un curso llamado tope indicará la posición actual del primer elemento de la pila.

Ejemplo de operaciones con pilas

```
void PilaLineal::insertar(TipoDeDato elemento)
{
    if (pilaLlena())
    {
        throw "Desbordamiento pila";
    }
    //incrementar puntero cima y copia elemento
    cima++;
    listaPila[cima] = elemento;
}
```

Referencias

-Aho, A., Hopcroft, J., Ullman, Jeffrey. (1998). *Estructura de datos y algoritmos (2da edición)*. México: S.A Alhambra Mexicana.

-Joyanes, L., Sánchez, L., Zahonero, I., (2007). *Estructura de datos en C++*. McGraw Hill