Implementación de pila para números complejos

En este proyecto, se busca dar a conocer la implementación de una pila aplicada para números complejos en lenguaje C.

¿Qué es una pila?

Una pila es una estructura de datos que permite almacenar y recuperar datos en la que el último dato en entrar es el primero en salir (Véase Imagen 1).

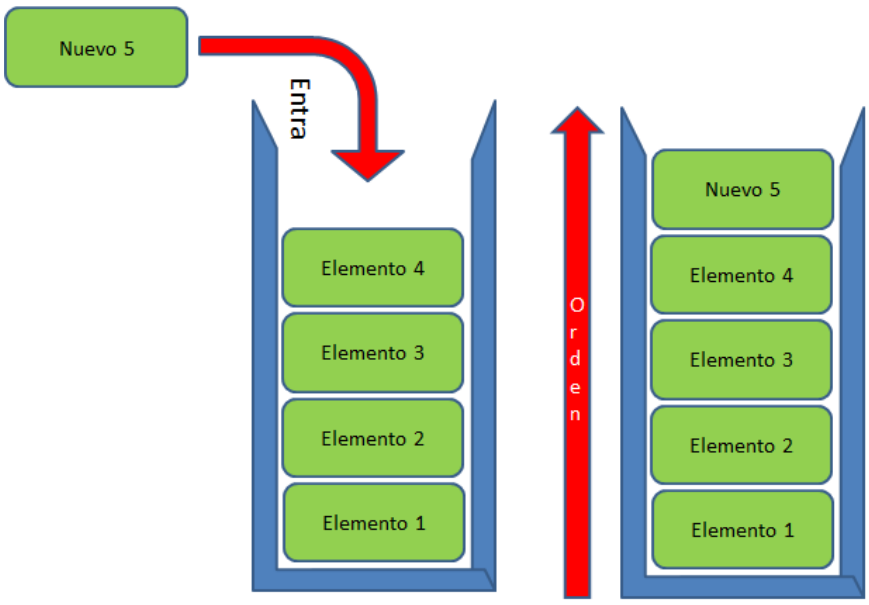


Imagen 1: Se muestra como el nuevo elemento que se agregue va en la parte de arriba, y si se quisiera sacar un elemento, el último que se agregó, sería el que se saldría.

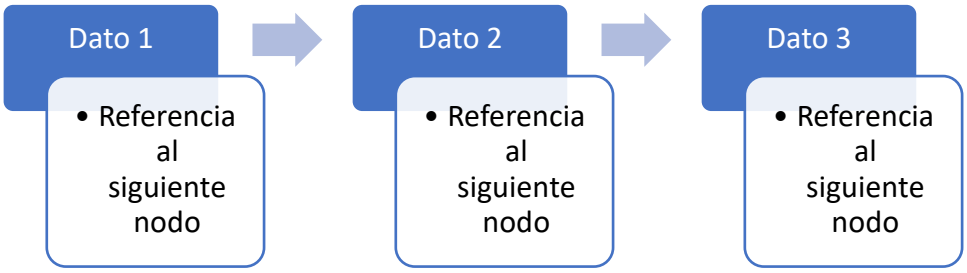
¿Qué operaciones se pueden realizar?

En una pila, existen 3 tipos de operaciones:

- Pop: Elimina el elemento del tope de la pila, es decir, el elemento de hasta arriba de la pila.
- Push: Añade un elemento hasta arriba de la pila.
- Peek: Devuelve el último elemento de la pila (el que está hasta arriba).

¿Qué es una lista ligada y cómo se puede implementar con una pila?

Es una estructura de datos y puede ser usada para implementar otras estructuras de datos. Consiste en una secuencia de nodos, en los que se guardan datos y se hace referencia al nodo que sigue (un puntero).



Además de ocupar el clásico arreglo para implementar una pila, también se puede usar una lista ligada, ya que cada nodo de la lista tendrá un elemento de la pila.

- Ofrece un control absoluto de todo lo que sucede en el ordenador.
- Organización del trabajo con total libertad.

¿Cómo se aplica los conocimientos anteriores al proyecto?

Todo el proyecto está implementado en lenguaje C. Se utiliza una lista ligada para armar la pila y el dato que se ocupa es el de números complejos.

En el proyecto se define la estructura de número complejo (Véase Imagen 2), que consta de una parte real y una imaginaria. También se define la estructura de nodo (la cual conformará la lista ligada).

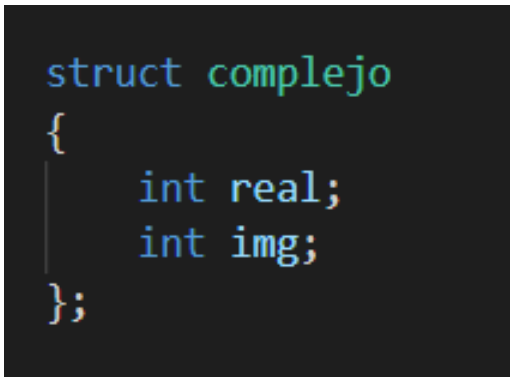


Imagen 2: Se muestra la implementación de una Estructura en C, en este caso, la estructura de número complejo

Se implementan los 3 tipos de operaciones descritas anteriormente. También se agrega una la cual recibe el nombre de “Is Empty” (Véase Imagen 3), la cual permite saber si la pila está vacía o no.

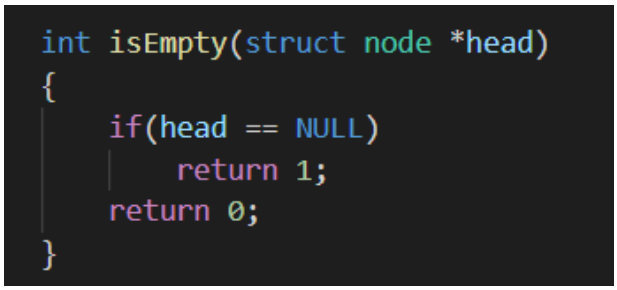


Imagen 3: Se muestra la función “is Empty”, en la cual si ya no hay un ningún elemento, regresa 1

Algo que se utiliza es el siguiente símbolo “->”, lo cual agrega un valor al atributo antes declarado en la estructura correspondiente (Véase Imagen 4).

```
struct node *push(struct node *head, struct complejo *dato)
{
    struct node* nuevo = (sizeof node*) malloc(sizeof *  node);
    if(nuevo == NULL)
    {
        exit(0);
    }

    nuevo->dato = dato;
    nuevo->siguiente = head;
    head = nuevo;
    return head;

}
```

Imagen 4: Se aprecia como se crea una estructura de tipo nodo (nuevo), y se le asigna a sus atributos (usando ->) dato y head.

Referencias

John Lewis, Joseph Chase. (2010). Java Software Structures: Designing and Using Data Structures. Addison-Wesley.