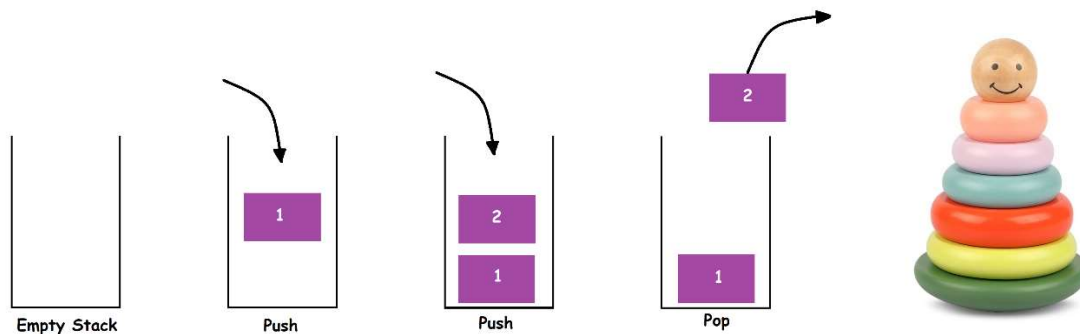


[A01] Generischer Stack auf Basis von verketteten Listen

Grafische Beschreibung



Textuelle Beschreibung

Das Stack arbeitet nach dem LIFO (Last In, First Out) Prinzip. Mittels der Methode `push()` wird ein Element auf den Stapel hinzugefügt und mittels `pop()` wird das oberste Element des Stapels entnommen.

Zu beachten gilt, dass nicht einfach ein beliebiges Element aus dem Stack entnommen werden kann, sondern die einzelnen Elemente, ausgehend von oben nach unten. Ein Beispiel hierbei finden wir bei dem Spielzeug „Regenbogenturm“.

Laufzeit

- `push()` -> $O(1)$
- `pop()` -> $O(1)$
- `size()` -> $O(1)$

Da im Grunde keine Datenverarbeitung stattfindet, besitzen die drei Methoden eine konstante Laufzeit $O(1)$. Es wird lediglich ein Wert zurückgeben bzw. ein neuer Wert/Pointer zugewiesen.

Wichtige Elemente

```
public T pop() throws StackEmptyException {  
  
    if (size == 0) // guard clause - to throw StackEmptyException  
        throw new StackEmptyException("There is nothing left for you :(");  
  
    T data = null; // create generic variable for data  
    data = first.getData(); // store data in variable from top of the stack  
    first = first.getNext(); // update pointer to -> "next in line"  
    size--; // reduce size of stack  
  
    return data;  
}  
  
public void push(T element) {  
    Node<T> node = new Node<>(element); // create new element for stack  
    Node<T> tmp = first; // store current top of stack as temporary node  
    first = node; // assign new node as first -> update it  
    first.setNext(tmp); // set reference to previous top -> next to pop  
    tmp = null;  
    size++; // increase stack size  
}
```