## Stack essentials

A **stack** is an abstract data type where elements are inserted and removed according to the **last-in-first-out (LIFO)** principle. The **push** operation inserts an item at the top of the stack, the **pop** operation removes the top item from the stack. Access to arbitrary elements is restricted. As a rule, a stack also supports the **peek** operation that just returns the current top element.

The underlying data structure to implement a stack can be an array or a linked list with restricted access to its elements.

In programming, stacks are used to:

- evaluate arithmetic expressions;
- store arguments of functions and results of the functions' calls;
- reverse the order of elements.

**The efficiency of stacks**
If you use a linked list or a classic array (non-resizable) as an internal structure, both **push** and **pop** operations always take constant O(1) time. It does not depend on how many elements there are in the stack, so the operations are very quick.