

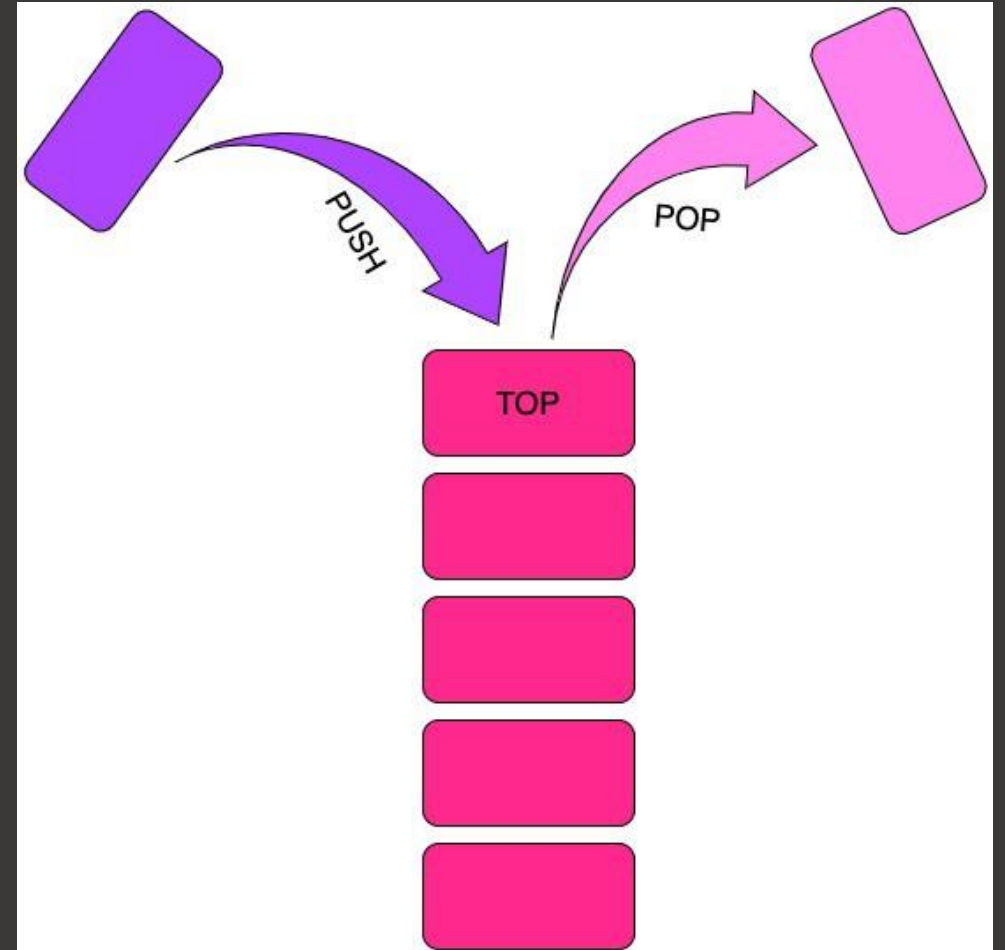
The background is dark gray with several concentric circles of varying line styles (solid and dashed) in a lighter gray. A white downward-pointing triangle is positioned to the left of the main title.

▼ Структура даних стек

Маслова Надія, ФВЕ 1 к. маг.

Стек

- Стек – абстрактний тип даних, що є списком елементів, організованих за принципом LIFO (last-in first-out останнім прийшов – останнім вийшов)
- Можна уявляти собі як стопку книжок або тарілок, де можна покласти один предмет згори і дістати верхній



Методи стеку

- Базові методи, які має підтримувати стек:

1. `push` – додати елемент
2. `top` – дістати верхній елемент
3. `pop` – видалити останній елемент

Додатково можливі методи `isEmpty`, `isFull`

Час роботи основних операцій складає $O(1)$

Реалізація (з використанням масиву)

```
#include<iostream>

using namespace std;

class Stack
{
public:
    int top_ind;
    int max = 1000;
    int myStack[max];
    Stack() { top = -1; }
    bool push(int x);
    void pop();
    int top();
    bool isEmpty();
};

bool Stack::push(int item)
{
    if (top_ind >= (max-1)) {
        cout << "Stack Overflow!!!";

        return false;
    }
    else {
        myStack[++top_ind] = item;
        return true;
    }
}

void Stack::pop(){
    if (top_ind < 0)    cout << "Stack Underflow!!";
    else    top_ind--;
}

int Stack::top(){
    int item = myStack[top_ind];
    return item;}

bool Stack::isEmpty(){
    return (top_ind < 0);}
```

Застосування

- Застосовується в задачах: збереження правильного порядку дужок у виразі; постфіксна, префіксна та інфіксна форма запису та перехід між ними; пошук з вертанням (backtracking)
- В сучасних комп'ютерах машинний стек використовується для потреб програми, що виконується: збереження змінних та визову підпрограм
- Використовується для обходу дерев та графів