**Лабораторная работа №8**

**«Стек»**

Стек – это структура данных, в которую вы можете добавлять данные в прямом порядке, а извлекать уже в обратном. Так называемый принцип LIFO = Last In First Out.

Не путайте стек – структуру данных и стек – область память, хотя работают они по схожему принципу ([(1) КАК РАБОТАЕТ СТЕК | ОСНОВЫ ПРОГРАММИРОВАНИЯ - YouTube](https://www.youtube.com/watch?v=MXoMuymbfo8)).

Подробнее: [Структуры данных и алгоритмы: стек | by Андрей Шагин | NOP::Nuances of Programming | Medium](https://medium.com/nuances-of-programming/%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D1%8B-%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85-%D0%B8-%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D1%8B-%D1%81%D1%82%D0%B5%D0%BA-fee8792751d2)

Вашей задачей будет создать шаблонный класс для реализации стека. В нем нужно предусмотреть методы:

1. Конструктор с указанием размера стека
2. Метод size
3. Метод empty – возвращает true или false
4. Метод push, чтобы положить в элемент в стек
5. Метод pop, чтобы извлечь элемент из стека
6. Метод top, чтобы посмотреть верхний элемент стека, но не извлекать его.

Добавьте обработку некорректных ситуаций через исключения:

1. Невозможно выделить необходимый участок памяти под стек (std::bad\_alloc).
2. Невозможно добавить новый элемент в стек (например, std::overflow\_error)
3. Невозможно извлечь элемент из стека, так как стек пустой (например, std::out\_of\_range)
4. Невозможно посмотреть элемент на вершине стека, так как стек пустой (например, std::logic\_error).

Продемонстрируйте работу вашего стека.