

## Лабораторная работа № 9. Фундаментальные структуры данных (стек, очередь, дек, список)

### Цель работы:

- 1 Изучить основы разработки модульных тестов.
- 2 Получить навыки работы с библиотекой GTest.

### Указание.

1. Разработать библиотеку классов для работы со структурами данных: «однонаправленный линейный список», «двунаправленный линейный список», «стек», «очередь», «дек»

### Задание 1. Стек

Создать шаблонные классы для реализации Стекa 1) на массиве `StackOnArray<T>` и 2) на последовательности связанных компонент `StackOnList<T>`, и публичную шаблонную структуру `NodeStack<T>`.

Структура `NodeStack` должна содержать:

- Публичное поле `value` типа `T` со значением, хранимым в данном элементе списка.
- Приватные поля для осуществления связи элементов в списке.
- Приватный конструктор (ничто кроме `StackOnList` не должно иметь возможности создания элементов типа `Node`).

Классы `StackOnArray<T>` и `StackOnList<T>` должны иметь следующий публичный интерфейс:

- Конструктор по умолчанию.
- Конструктор от `std::initializer_list<T>`.
- Методы `Size()` и `IsEmpty()`.
- Метод `Push(const T& value)` осуществляющие вставку элемента в вершину стека.
- Метод `NodeStack* Top()`, позволяющие получить первый элемент в стеке, не извлекая его.
- Метод `T Pop()`, осуществляющие извлечение первого элемента из стека.
- Проверить корректность вызовов и передаваемых параметров с помощью `assertов`.
- Создать константный аналог для метода `Top`, который будет возвращать `const NodeStack*`.
- Обеспечить возможность проверки на равенство/различие стеков при помощи соответствующих операторов.
- Реализовать конструктор копирования и копирующий оператор присваивания.
- Реализовать конструктор перемещения и перемещающий оператор присваивания.

Подумайте, какая проблема существует в описанном интерфейсе, объясните её и придумайте, как можно было бы её избежать.

### Задание 2. Использование стека для решения задач. Вычислить значение формулы с *min*, *max*

Пусть дана без ошибок формула, имеющая следующий синтаксис:

`<формула>::=<цифра> | max(<формула>, < формула >) | min (<формула>,< формула >)`

`<цифра>::=0 1 2 3 4 5 6 7 8 9`

Например, 8 или `max(4, min(3, 5))` или `min(min(3, 5), max(2, max(3, 4)))`.

Вычислить значение формулы, используя структуру данных стек.

**Задание 3. Использование стека для решения задач. Вычислить значение выражения**

Дана строка, которая представляет собой исходное математическое выражение, содержащее целые числа (*\* можно рассматривать и дробные числа*), скобки любой степени вложенности, и операции:

Приоритет	Операция	Описание
<b>5</b>	* / %	Умножение, деление и остаток
<b>6</b>	+ -	Сложение и вычитание
<b>7</b>	<< >>	Побитовый сдвиг влево и вправо
<b>8</b>	< <=	Операции сравнения < и ≤
	> >=	Операции сравнения > и ≥
<b>9</b>	== !=	Операции сравнения = и ≠
<b>10</b>	&	Побитовое И
<b>11</b>	^	Побитовый XOR (исключающее ИЛИ)
<b>12</b>		Побитовое ИЛИ (inclusive or)
<b>13</b>	&&	Логическое И
<b>14</b>		Логическое ИЛИ

(Приоритет операций в C++ : <https://ru.cppreference.com/w/cpp/language/operator%20precedence>)

1. Перевести его в обратную польскую запись – построить выражение, в постфиксной записи, без скобок.
2. Вычислить значение выражения в постфиксной записи.

**Задание 3. Тестирование реализаций стека и задач.**

Разработать тесты для тестирования функционала ваших приложений.