Лабораторная работа №7

В рамках данной лабораторной работы вам предстоит написать собственный **класс Vector**, который будет являться упрощённым аналогом шаблонного класса **std::vector**. В этой лабораторной запрещено использовать стандартные контейнеры и умные указатели, поэтому *аккуратно работайте с памятью.*

В классе **Vector** будем хранить элементы типа **int**, а также в нём есть:

- Конструктор по умолчанию.
- Конструктор, принимающий размер вектора и заполняющий его нулями.
- Конструктор, принимающий **std::intitializer_list<int>** — это позволит создавать вектор следующим образом:

```
Vector v{1, 2, 3, 4, 5};
```

Листинг 1.1: Создание объекта класса Vector с помощью std::initializer_list

- Конструктор копирования.
- Оператор присваивания копированием.
- Деструктор.
- Метод ***Swap***, который принимает другой вектор по ссылке и меняет содержимое текущего вектора с ним.
- Операторы индексирования: константный и нет. Последний должен позволять *менять содержимое контейнера по индексу.*
- Метод ***At***, который работает аналогично оператору индексирования (т.е. у него тоже будет две версии: константная и нет). Отличие этого метода в том, что он проверяет *выход за границу массива* — в этом случае выбрасывается исключение ***std::out_of_range***.
- Метод ***Size***, возвращающий число элементов в контейнере.
- Метод ***Capacity***, возвращающий текущее число выделенных ячеек памяти под вектор.

- Метод ***PushBack***, который добавляет элемент в конец вектора. Если при этом память, которая выделена для вектора, заполнена, то выполните *реаллокацию*: выделите массив вдвое большего размера, скопируйте (или переместите) туда элементы, после чего удалите старый массив. В этом случае **capacity** также должна увеличится вдвое.

- Метод ***PopBack***, который удаляет последний элемент вектора. Сужать вектор при этом не нужно, должен измениться только **size**.

- Метод ***Clear***, который делает контейнер пустым. Аналогично, сужать вектор при этом не нужно, **size** должен стать нулевым.

- Метод ***Reserve***, принимающий новое значение **capacity**, что позволяет зарезервировать место в векторе. Если текущая **capacity** не мешьне заданного, то метод ничего не должен делать. В ином случае выполните реаллокацию в массив размера **capacity**.

- Оператор вывода. Оператор вывода должен отображать вектор следующим образом:

```
Vector v{1, 2, 3, 4, 5};
std::cout << v; // вывод на консоль [1, 2, 3, 4, 5]
Vector v_empty;
std::cout << v_empty; // вывод на консоль []
```

Листинг 1.2: Пример вывода вектора на консоль

Обратите внимание, что оператор вывода *не являются членами класса*, а определяются вне класса как обычные функции.