**Слуцкий Никита. 053501. Лабораторная Работа №7. Реализация класса Deque для int.**

В основе класса лежит работа с указателями. Класс Дек представляет из себя односвязный список (list), в котором каждый член (Node, который также выделен в отдельный класс) содержит 2 поля: value (непосредственно значение) и \*next (указатель на следующий после него элемент).

**В классе присутствуют следующие поля:**

* \*first – хранит адрес текущего первого элемента дека
* \*last – соответственно хранит адрес последнего
* \*length – текущее количество элементов дека

**Реализованы 2 вида конструктора. Можно объявить:**

* Новый пустой дек
* Дек на основе уже имеющегося (произойдёт копирование)

Реализован деструктор, который вызывает метод RemoveAll() и удаляет все Node

Реализованы следующие методы:

* Size ( ) – возвращает значение приватного поля length
* FindFirst ( ) – линейно ищет элемент в деке и возвращает его индекс (в случае отсутствия: -1)
* Push ( ) - добавляет элемент в конец дека. Так как присутствует поле \*last, работает за константное количество операций (иначе необходимо было бы вначале линейно пройтись от \*first до последнего, а потом уже добавлять)
* Unshift ( ) – добавляет элемент в начало дека. Название по аналогии с JavaScript, где метод делает то же самое для массивов
* Insert ( ) – вставляет в указанную позицию переданный элемент. В худшем случае работает линейно за O (N ). Так как, опять же, нужно искать элемент последовательно начиная с первого (так как только предыдущий знает о следующем). Если бы была двусвязная реализация, можно было бы свести к O (N / 2)
* Pop ( ) – удаляет элемент с конца (на самом деле сводится к Remove(length -1))
* Shift ( ) – удаляет элемент с начала
* Remove ( ) – удаляет элемент по указанному индексу
* RemoveAll ( ) – полностью очищает дек

Перегружены некоторые операторы:

* Оператор потокового вывода cout<<. По аналогии в JavaScript, где можно вывести массив ( console.log(array) ), он выводит размер и элементы в скобочках
* Оператор доступа по индексу [ ] (как и в обычном массиве или векторе)
* Оператор присваивания. Можно не копировать вектора через конструктор, а присваивать