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

В основе вектора лежит динамический массив int.

Работаю по следующему принципу: при объявлении выделяется «стартовая память» на 2 элемента; каждый раз при Push( ) если память кончается, выделяется новый кусок памяти, который в 2 раза больше предыдущего, и старое содержимое туда копируется. Таким образом каждые Log(N) необходимо переносить N элементов. Это не сильно портит асимптотику.

**В классе присутствуют следующие поля:**

* length – текущее количество элементов в векторе
* realSize – текущий выделенный размер динамического массива
* arr – непосредственно динамический массив

**Реализованы 4 вида конструктора для разных передаваемых параметров. Можно объявить:**

* пустой вектор из нулей;
* вектор с заданной длиной из нулей;
* вектор с заданной длиной и стартовым заполнением;
* вектор на основе уже имеющегося (произойдёт копирование);

**Реализован деструктор, который освобождает память от динамического массива.**

**Реализованы следующие методы (приватные и публичные, в зависимости от надобности):**

* CopyArrays ( ) – для копирования старого содержимого в новый массив при переходе «на новую размерность»
* Size ( ) – возвращает текущее количество элементов в массиве
* Print ( ) – выводит элементы через пробел (но ниже переопределён cout<<)
* Top ( ) – возвращает последний элемент в текущем векторе
* FindFirst ( ) – линейно ищет элемент в векторе и возвращает его индекс (или код ошибки, если не нашёлся)
* Push ( ) – вставляет элемент в конец вектора
* Insert ( ) – вставляет в указанную позицию указанное число
* Pop ( ) – удаляет последний элемент вектора
* Remove ( ) – удаляет указанный (по индексу) элемент
* RemoveAll ( ) – полностью очищает вектор. Сводится просто к очистке памяти динамического массива, выставления realSize по умолчанию на 2, length на 0

**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