**Курс «Объектно-ориентированное программирование на C++»**

**Встреча №16**

Тема: динамические структуры данных – двусвязный список

Домашнее задание

**Задание 1.** Реализовать шаблонный класс "Очередь" на основе двусвязного списка.

**Задание 2.** В существующий класс двусвязного списка добавить: операцию клонирования списка (функция должна возвращать адрес головы клонированного списка), перегрузить оператор + (оператор должен возвращать адрес головы нового списка, содержащего элементы обоих списков для которых вызывался оператор), перегрузить оператор \* (оператор должен возвращать адрес головы нового списка, содержащего только общие элементы обоих списков для которых вызывался оператор).

**Задание 3.** Создать шаблонный класс-контейнер Array, который представляет собой массив, позволяющий хранить объекты заданного типа. Класс должен быть реализован с помощью **двусвязного списка**. Класс должен реализовывать следующие функции:

1. GetSize - получение размера массива (количество элементов, под которые выделена память)
2. SetSize(int size, int grow = 1) - установка размера массива (если параметр size больше предыдущего размера массива, то выделяется дополнительный блок памяти, если нет, то "лишние" элементы теряются и память освобождается); параметр grow определяет для какого количества элементов необходимо выделить память, если количество элементов превосходит текущий размер массива. Например, SetSize(5, 5); означает, что при добавлении 6-го элемента размер массива становится равным 10, при добавлении 11-го - 15 и т. д.
3. GetUpperBound - получение последнего допустимого индекса в массиве. Например, если при размере массива 10, вы добавляете в него 4 элемента, то функция вернет 3.
4. IsEmpty - массив пуст?
5. FreeExtra - удалить "лишнюю" память (выше последнего допустимого индекса)
6. RemoveAll - удалить все
7. GetAt -получение определенного элемента (по индексу)
8. SetAt - установка нового значения для определенного элемента (индекс элемента должен быть меньше текущего размера массива)
9. operator [] - для реализации двух предыдущих функций
10. Add - добавление элемента в массив (при необходимости массив увеличивается на значение grow функции SetSize)
11. Append - "сложение" двух массивов
12. operator =
13. GetData - получения адреса массива с данными
14. InsertAt - вставка элемента(-ов) в заданную позицию
15. RemoveAt - удаление элемента(-ов) с заданной позиции