1.Разработать класс "последовательность" в соответствии со следующим заданием:

Состояние класса -

Последовательность состоит из элементов типа ***int*** и определяется размером (количество элементов в последовательности) и значениями своих элементов, например: последовательность из 6 элементов {12, 8, 5, -3, 11, 5}. Память под последовательность выделяется статически и задается массивом фиксированного предельного значения (например, максимальный размер последовательности 100 элементов).

Протокол класса -

Определяет возможности создания и инициализации экземпляров класса и правила использования их (методы класса).

Предусмотреть следующие возможности:

·          пустой конструктор для инициализации экземпляров и массивов экземпляров класса по умолчанию;

·          создание экземпляров класса с инициализацией единственным элементом последовательности;

·          создание экземпляров класса с инициализацией размером и значениями элементов последовательности;

·          ввод экземпляров класса из входного потока и вывод их значений в выходной поток;(???????)

·          объединение двух последовательностей в одну; в результате формируется новый вектор с сохранением исходных(перегрузка?)(+)

·          выделение первой упорядоченной по возрастанию или по убыванию последовательности длиной не менее трёх элементов;(оператор вызова функции)

·          выполнение операции включения в последовательность нового элемента;(+=)

·          определение количества групп в последовательности (то есть, сколько разных элементов включено в последовательность);

·          определение частоты появления некоторого элемента в последовательности (то есть, сколько раз заданный элемент встретился в последовательности).

2. Проектирование класса рекомендуется начать с представления состояния класса, учитывающего заданные операции, а затем реализации конструкторов и метода вывода. Для отладки и исчерпывающего тестирования других методов разработанного класса реализовать диалоговую программу, которая позволяет вводить параметры отлаживаемых методов. Для обработки ошибочных ситуаций использовать механизм исключительных ситуаций.

-------------------------------------------------------------------------------------------------------------------------------------------------------------

3. Повторить разработку класса, реализовав отдельные методы (там, где это оправданно), перегруженными операторами. Состав перегруженных операторов целесообразно согласовать с преподавателем.

4. Еще раз повторить разработку класса при условии, что память под массив необходимой длины выделяется динамически, во время выполнения программы (с помощью оператора ***new***; память задается указателем на ***char*** в состоянии класса). Для хранения данных в экземпляре класса не должна использоваться лишняя память.

Дополнить интерфейс класса следующими возможностями:

·          создание экземпляра класса с его инициализацией другим экземпляром класса (копирующий конструктор);

·          переопределение экземпляра класса (с помощью перегруженного оператора присваивания).

Разработать и реализовать прикладную программу, использующую класс, разработанный другим студентом. Задание для прикладной программы разработать самостоятельно и согласовать с преподавателем.