**Лабораторная работа №7**

* Создать класс для работы с однонаправленным линейным списком.
* Элементы списка реализовать как экземпляры класса.
* Класс должен содержать переопределенный метод ToString() и реализовывать интерфейс IEnumerable.
* Реализовать следующие методы этого класса однонаправленный список:

1. Добавление элемента в начало списка
2. Добавление элемента в конец списка
3. Вывод списка на экран
4. Поиск элемента по его значению (возвращает bool)
5. Поиск элемента по номеру (возвращает значение)
6. Добавление элемента перед заданным
7. Добавление элемента после заданного
8. Удаление элемента из начала списка
9. Удаление элемента из конца списка
10. Удаление элемента перед заданным

11.Удаление элемента после заданного

* Организовать текстовое меню, содержащее все пункты задания.
* Задания по вариантам реализовать как методы класса Program:

1. Вставить в список новый элемент Е1 перед первым вхождением элемента Е
2. Перенести в конец непустого списка его первый элемент
3. Вставить в непустой список новые элементы Е1 и Е2 перед последним элементом списка
4. Удалить из списка все отрицательные элементы
5. Перевернуть список, т.е. изменить ссылки в этом списке так, чтобы его элементы оказались расположенными в обратном порядке
6. Удалить из списка второй элемент, если такой есть
7. Проверить, есть ли в списке хотя бы два одинаковых элемента
8. Удалить из списка первый отрицательный элемент, если такой есть
9. Вставить в список новый элемент Е1 перед каждым вхождением элемента Е
10. Оставить в списке только первые вхождения одинаковых элементов
11. Перенести в начало списка его последний элемент 12.Продублировать каждый элемент списка
12. Вставить в середину непустого списка новый элемент Е1, если количество элементов списка четно.
13. Удалить из непустого списка за каждым вхождением элемента Е один элемент, если такой есть и он отличен от Е.
14. Вставить после каждого элемента, занимающего четную позицию в списке, новый элемент Е1
15. Удалить из непустого списка за каждым вхождением элемента Е один

элемент, если такой есть.

1. Подсчитать количество элементов списка, имеющих равных соседей. Под соседями понимаются элементы, находящиеся перед и после текущего элемента.
2. Переставить в обратном порядке элементы, находящиеся перед первым и последним вхождениями элемента Е.
3. Добавить в конец списка все его элементы в обратном порядке. 20.Удалить элементы списка, имеющие равных соседей. Под соседями

понимаются элементы, находящиеся перед и после текущего элемента. 21.В списке из каждой группы подряд идущих элементов оставить только

один

1. Удалить из непустого списка перед каждым вхождением элемента Е один элемент, если такой есть.
2. Найти максимальный элемент списка
3. Вставить в непустой список, элементы которого упорядочены по возрастанию, новый элемент так, чтобы сохранилась упорядоченность
4. Удалить из непустого списка перед каждым вхождением элемента Е один элемент, если такой есть и он отличен от Е.
5. Поменять местами первый и последний элементы списка 27.Проверить, упорядочены ли элементы списка по возрастанию 28.Проверить на равенство списки L1 и L2

29.Проверить, входит ли список L1 в список L2 30.Добавить в конец списка L1 и все элементы списка L2

1. Вставить в список L1 за первым вхождением элемента E (если такой есть) все элементы списка L2
2. Создать новый список L, содержащий элементы списков L1 и L2, чередующиеся между собой (если списки L1 и L2 одинаковой длины)
3. Отсортировать элементы списка по возрастанию значений информационного поля
4. Создать новый список L, содержащий элементы отсортированных по возрастанию списков L1 и L2 с сохранением порядка следования элементов.
5. Сформировать новый список L, включив в него по разу элементы, которые входят в один из списков L1 или L2 .
6. Сформировать новый список L, включив в него по разу элементы, которые входят в оба списка L1 и L2 .
7. Сформировать новый список L, включив в него по разу элементы, которые входят в список L1 и не входят в список L2 .