

## Задание 2 для первого семестра 2 курса

Требуется разработать и реализовать параметризованный класс списочного типа. Класс должен содержать конструктор без параметров, конструктор копирования. Набор методов класса должен соответствовать идеологии работы со списками, т.е. должно быть введено понятие текущей позиции и реализованы функции перемещения этой позиции и доступа и изменения списка в окрестности текущей позиции.

В классе должны быть реализованы оператор присваивания и предусмотрено переопределение одной-двух операций по смыслу задачи (например, сложение, как объединение множеств данных). В классе должна быть реализована операция сортировки хранящегося набора данных, а также распечатки всего набора данных в файл или на экран в наглядной форме.

Контроль за возможностью доступа к хранимым значениям и за другими возможными отказами выполняется через механизм исключений (throw — try-catch)

Релизация тестируется для двух-трех различных типов хранимых объектов — простого и “сложного”.

### Варианты структур данных

- А. Однонаправленный список
- В. Двухнаправленный список
- С. Кольцевой однонаправленный список
- Д. Кольцевой двухнаправленный список
- Е. Динамический массив как список блоков
- Ф. Стек на базе списка
- Г. Очередь на базе списка
- Н. Дек на базе кольцевого списка

### Типы объектов

Работа параметризованной реализации проверяется и тестируется на различных типах данных:

1. базовые типы (числа `int`, `double`) классы из своего первого задания
2. простейшие структурные типы (точка, вектор  $R^3$ )
3. сложные классы — классы из своего первого задания.

Для тестирования в списочном классе нужно предусмотреть параметризованную функцию распечатки содержимого списка, и соответственно реализовать функции распечатки для каждого конкретного используемого типа.

Работа над заданием предполагает еженедельный контроль по следующему графику (естественно, его можно выполнять с опережением)

### График выполнения задания

неделя 1: разработка и согласование описания (интерфейса) класса.

неделя 2: реализация и первичный тест на простых типах данных, проверка работы механизма исключений

неделя 3: тест на сложных типах данных

недели 4: доработка, исправление и окончательная сдача задания