# Общее задание

1. Разработать интерфейс `ICollection`, от которого будут порождены различные структуры для хранения элементов типа `Type`. Под интерфейсом понимают абстрактный класс, у которого все методы виртуальные. Тип `Type` — это псевдоним, который следует объявить через **`typedef`**, например, для **`int`** или **`double`**. У интерфейса `ICollection` имеются следующие методы:
   а) `Type at(int index)` — возвращает содержимое элемента коллекции, индекс которого равен `index`;
   б) `Type operator[] (int index)` — возвращает содержимое элемента коллекции, индекс которого равен `index`;
   в) `void insert(size_t index, const Type& element)` — вставляет в коллекцию элемент element в позицию `index`;
   г) `void removeAt(size_t index)` — удаляет элемент из коллекции элемент, порядковый номер, которого равен `index`;
   д) `Type getFirst()` — возвращает первый элемент коллекции;
   е) `Type getLast()` — возвращает последний элемент коллекции.
2. Создать классы `SinglyLinkedList` (односвязный список) и `DoublyLinkedList` (двусвязный список), которые должны поддерживать интерфейс `ICollection`. Дополнить данные классы методом `append()`, который добавляет элемент в конец списка.
3. Написать функцию, которая осуществляет пузырьковую сортировку элементов коллекции. Апробировать её работу с объектами классов `SinglyLinkedList` и `DoublyLinkedList`.
4. Разработать класс очереди `Queue`. Данный класс должен использовать в своей работе класс `DoublyLinkedList`, для этого использовать закрытое наследование. У очереди есть метод добавления элемента в конец `enqueue()` и метод извлечения элемента с начала очереди `extract()`. Размер очереди устанавливается на этапе создания объекта (является параметром конструктора). В случаях, когда выполняется добавление элемента в очередь, когда она уже заполнена, или извлечение первого элемента из очереди, когда в ней нет элементов, генерируются исключения. Апробировать работу очереди.

5 Разработать класс стека `Stack` и апробировать его работу. Данный класс должен использовать в своей работе класс `SinglyLinkedList`, для этого объект класса `SinglyLinkedList` должен быть полем объекта класса `Stack`. У класса должны быть методы:

а) `push()`, помещающий элемент на вершину стека;

б) `pop()`, извлекающий элемент с вершины стека.

Наибольший размер стека устанавливается на этапе создания объекта (является параметром конструктора). В случаях, когда выполняется добавление элемента в уже полный стек и извлечение элемента из пустого стека, генерируются исключения.

Обработку ошибок следует реализовать при помощи исключений. В качестве класса для объекта-исключения подойдут `std::logic_error` и `std::out_of_range` из заголовочного файла `<stdexcept>`. Допускается использовать и собственные классы-исключения.