

Домашнее задание № 1

Задача 1

Класс, представляющий простой односвязный список, описан следующим образом:

```
public class List<E extends Comparable<E>> {  
    private static class Node<E> {  
        E info;  
        Node<E> next;  
    }  
    Node<E> head = null;  
    ...  
}
```

Требуется дополнить определение класса двумя публичными методами:

1. **public void swap(int i, int j)** { ... } - переставляет в списке местами два элемента с номерами *i* и *j*. Нумерация элементов списка - начиная с нуля. Номера могут идти в любом порядке, то есть может быть как $i < j$, так и $j < i$. Если $i == j$, то список не меняется. Если хотя бы один из индексов выходит за границу списка (меньше нуля или больше или равен числу элементов списка), то должно возникать прерывание `IndexOutOfBoundsException`.
2. **public void sort()** { ... } - упорядочивает элементы списка в порядке возрастания (в соответствии с порядком, определенным интерфейсом `Comparable`). Не нужно изобретать сложные алгоритмы сортировки, для списков самым подходящим будет алгоритм простых вставок.

В обоих методах не надо создавать копии узлов списка, все манипуляции должны происходить только с указателями.

Содержимое узлов менять нельзя! Перестановка элементов в списке осуществляется только за счет того, что меняются ссылки из одних узлов на другие.

Задача 2

Интерфейс, представляющий простой односвязный циклический список, описан следующим образом:

```
public interface CycList<E> extends Iterable<E>{  
    void shift(int delta);  
}
```

Функция `shift` должна сдвигать указатель циклического списка на заданное количество элементов вперед по списку.

Требуется реализовать этот интерфейс и написать еще одну функцию (не метод класса!), которая, используя заданный интерфейс, проверяет, совпадают ли два заданных списка покомпонентно, то есть содержат ли они одни и те же элементы с точностью до сдвига. Например, если списки содержат элементы `[1, 2, 3, 4, 5]` и `[3, 4, 5, 1, 2]`, то они равны, а если `[1, 2, 3, 4, 5]` и `[5, 4, 3, 2, 1]`, - то не равны. Равенство элементов списков понимается в смысле метода `equals`.

Функция должна иметь следующий заголовок:

```
static <E> boolean listsEqual(CycList<E> list1, CycList<E> list2) { ... }
```

Расширять заданный интерфейс дополнительными функциями нельзя, но в реализации, конечно, можно добавлять какие угодно функции для создания списка, вывода его элементов и т.п.