**Задачи на тему: Параметризация, Java – коллекции, часть I**

**1. Реализовать связный список, представленный на UML диаграмме ниже.**

**C:\Users\GLEB\Downloads\Untitled Diagram.png**

**Класс MyLinkedList реализует интерфейс ILinkedList, содержащий в себе описание основных операций, для работы со связным списком. Данные операции полностью аналогичны, операциями в Java реализации. Описание можно посмотреть в Java документации (например, здесь** [**https://docs.oracle.com/javase/7/docs/api/java/util/List.html**](https://docs.oracle.com/javase/7/docs/api/java/util/List.html)**)**

**Подробную информацию о связных списках можно найти на просторах интернета.**

**Реализацию класса MyLinkedList продумываем самостоятельно. Можно вводить любые вспомогательные переменные и методы на ваше усмотрение. В качестве реализации поведения итератора используем родной Java интерфейс -** [**https://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html**](https://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html)

**Создаём объекты итератора любыми известными вам способами. Например, можно создать отдельный класс, реализующий интерфейс Iterator, можно создать фабрику, можно предоставить анонимную реализацию итератора и т.д.**

**Тестируем все операции в классе MainClass. В указанный класс добавляем метод main и тестируем в нём все операции. В узлах располагаются объекты любого типа. Можно использовать классы из предыдущих заданий. Например, Circle, MyTriangle и т.д.**

**Также будет интересно сравнить производительность, написанной вами коллекции, с Java реализацией.☺ Для этого необходимо создать объекты MyLinkedList и java.util.LinkedList. Используя System.nanoTime оценить примерно время выполнения основных операций – добавления, поиска, удаления элемента. Количество элементов в коллекции подбираем самостоятельно. Возьмите для начала 10000 элементов. При необходимости количество элементов можно варьировать на ваше усмотрение.**

**2. Сравнительный анализ производительности основных Java коллекций.**

**Написать тестовый класс, измеряющий с помощью System.nanoTime() время выполнения основных операций – добавление, вставка, удаление элементов**

1. **Сравнить производительность ArrayList и LinkedList для основных операций**
2. **Сравнить производительность HashSet, LinkedHashSet, TreeSet для основных операций**
3. **Сравнить производительность HashMap, LinkedHashMap, TreeMap**

**Размер, тип, количество тестовых данных выбираем самостоятельно. Это уже ваше творчество ☺ Но количество объектов в коллекции не менее 10000, а при необходимости увеличить до 100000**

**На основании полученных вами результатов пришлёте мне вместе с кодом, выводы☺**

**В выводе необходимо указать в каких случаях удобно использовать ту или иную коллекцию (для каждого из пунктов a,b,c)**