# Задание

1. В программу демонстрации классов, созданную в лабораторной работе №2, включить класс (или метод), создающий коллекцию типа **ArrayList** для объектов главного класса\* примера и (на выбор) коллекцию типа **LinkedList** или **HashMap** того же класса. Т.е. реализовать два варианта коллекции объектов.
2. Для заполнения коллекций реализовать класс или метод генерации случайных значений для заполнения объектов коллекций.
3. Процесс заполнения коллекций записывать в лог в формате: **тип операции, ключ (индекс), время выполнения операции (в миллисекундах или наносекундах)**. Варианты типа операции: **add** — добавление (**put** для **HashMap**), и (на выбор) **remove** - удаление или **set** - изменение (**replace** для **HashMap**). В конце лог-файла указывается общее вемя выполнения всех операций и среднее время выполнения операции определенного типа.

Пример записи лога работы ArrayList для 10-ти значений:

Start program: 11.10.2016 16:10:21

ArrayList

add, ID = 1, 14662984

add, ID = 2,1067426

...

add, ID = 9, 951959

add, ID = 10, 659870

addTotalCount = 10

addTotalTime = 53674775

addMedianTime = 659642

remove, ID = 4, 547399

removeTotalCount = 1

removeTotalTime = 547399

removeMedianTime = 547399

Finish program: 11.10.2016 16:10:23

Эксперименты проводятся для количества элементов в коллекциях: 10, 100, 1000, 10000, 100000. Удалять или изменять нужно 10% элементов из каждой группы (10 - 100000). Индексы элементов для удаления или изменения подбираютсся случайным образом (т.е. с помощью генератора случайных чисел). Время выполнения операции рассчитывается, как разность между текущим и предыдущим шагами выполнения операции. Для получения времени в миллисекундах можно использовать - **System.*currentTimeMillis*()**, а для получении времени в наносекундах - **System.nanoTime()**. **Итого должно получиться 10 логов:**5 для ArrayList (10, 100, 1000, 10000, 100000)  
5 для LinkedList или HashMap (10, 100, 1000, 10000, 100000)

1. Запись в лог и обработку ошибок реализовать через собственный класс обработки сообщений. Внутри класса должен быть реализован счетчик ошибок. Все сообщения об ошибках должны обрабатываться в данном классе.
2. Сделать выводы о степени регресса производительности при добавлении/удалении (или) изменении элементов в коллекциях разных типов. (Сравнивая значения времени в Total и Median)

**Дополнительное задание (+1 балл)**Для коллекции **ArrayList** проверить, заметно ли изменение времени добавления элемента при увеличении размера по формуле: **(oldCapacity \* 3) / 2 + 1**.

\* - главным классом, считаем класс, содержащий иерархию классов лабораторной работы №2. Например, если есть класс **Forest** (лес), содержащий в себе классы: **Fauna** (животные), **Flora** (растения), то главным классом будет **Forest** и нужно создать **ArrayList**, (**LinkedList** или **HashMap**), содержащий объекты типа **Forest**. И записать время исполнения операции добавления (**add | put**) объектов типа **Forest** в этот список, а затем время удаления (**remove**) или изменения (**set | replace**) 10% случайных элементов в списке.