**Цель задания**

Научиться сортировать и искать элементы в коллекциях.

**Что нужно сделать**

**1.** Напишите генератор «красивых» автомобильных номеров и методы поиска элементов в коллекциях:

* прямым перебором по ArrayList,
* бинарным поиском по сортированному ArrayList,
* поиском в HashSet,
* поиском в TreeSet.

**2.** Измерьте и сравните длительность каждого метода поиска.

Формат вывода результатов поиска:

Поиск перебором: номер найден/не найден, поиск занял 34нс

Бинарный поиск: номер найден/не найден, поиск занял 34нс

Поиск в HashSet: номер найден/не найден, поиск занял 34нс

Поиск в TreeSet: номер найден/не найден, поиск занял 34нс

**3.** Напишите в форме ответа, какой поиск — самый быстрый, а какой — самый медленный.

В видео [Поиск и сортировка](https://go.skillbox.ru/course/java-s-nulya/0f096167-e333-4c25-bf93-fadd13393e38) неточно указана оценка результата работы метода бинарного поиска Collections.binarySearch(). Метод возвращает int, если возвращаемое значение больше или равно нулю — это означает, что элемент найден. Если возвращаемое значение int меньше нуля — элемент в коллекции не найден.

**Рекомендации**

* Сортировка не входит в учёт времени для бинарного поиска.
* Для детального сравнения методов поиска используйте время в наносекундах:

     System.nanoTime()

* Используйте правила генерации номеров для получения более 2 млн номеров:

XYZ — различные буквы, N — цифры, R — регион (от 01 до 199);

XNNNYZR — *пример, A111BC197, Y777HC66*,

таким образом, количество номер будет достаточно для оценки времени поиска даже в миллисекундах.

**Критерии оценки**

«Зачёт» — при вводе в консоль автомобильного номера программа однозначно отвечает, найден ли номер в каждой из коллекций.  
«Незачёт» — задание не выполнено.