#### Цель задания

Научиться сортировать и искать элементы в коллекциях.

#### Что нужно сделать

Выполните задание в классе ****src/main/java/practice/CoolNumbers.java.****

1. Реализуйте метод генерации «красивых» автомобильных номеров generateCoolNumbers() в классе CoolNumbers. Используйте следующие правила генерации номеров:
   1. XYZ — различный набор из списка разрешенных букв, N — цифры, R — регион (от 01 до 199);
   2. XNNNYZR — пример: A111BC197, У777HC66.  
      В госномерах автомобилей в РФ используются следующие буквы: А, В, Е, К, М, Н, О, Р, С, Т, У, Х.  
      В коллекции должно быть не менее 2 млн номеров.
2. Реализуйте методы поиска класса CoolNumbers по поиску номеров в каждой коллекции:
   1. прямым перебором по ArrayList,
   2. бинарным поиском по сортированному ArrayList,
   3. поиском в HashSet,
   4. поиском в TreeSet.
3. Измерьте и выведите длительность каждого метода поиска. Формат вывода результатов поиска:
   1. Поиск перебором: номер найден/не найден, поиск занял 34нс
   2. Бинарный поиск: номер найден/не найден, поиск занял 34нс
   3. Поиск в HashSet: номер найден/не найден, поиск занял 34нс
   4. Поиск в TreeSet: номер найден/не найден, поиск занял 34нс

#### Рекомендации

* сортировка не входит в учёт времени для бинарного поиска;
* для детального сравнения методов поиска используйте время в наносекундах:   System.nanoTime()

#### Критерии оценки

****«Зачёт»**** — при вводе в консоль автомобильного номера программа однозначно отвечает, найден ли номер в каждой из коллекций. Все тесты успешно выполняются.

****«На доработку»**** — задание не выполнено.