|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| *Федеральное государственное бюджетное образовательное учреждение высшего образования* ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**Дисциплина «Программирование на языке Джава»**

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКИМ ЗАНЯТИЯМ №19, №20**

Выполнил студент группы ИНБО-02-20 Чернов Я. А.

Принял Степанов П.В.

Практические работы выполнены «\_\_\_»\_\_\_\_\_\_\_2021г.

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_»\_\_\_\_\_\_\_2021г.

Отметка о выполнении

**Москва 2021 г.**

**Практические занятия №19-20**

**Задание**

1. Напишите генератор «красивых» автомобильных номеров. Используйте правила генерации номеров для получения более 2 млн номеров:

* X, Y, Z — различные буквы автомобильного номера ("А", "В", "Е", "К", "М", "Н", "О", "Р", "С", "Т", "У", "Х"), N — цифры, R — регион (от 01 до 199);
* XNNNYZR — пример, A111BC197, Y777HC66

1. Используя генератор «красивых» методов проведите поиск номера, введенного из консоли, с помощью методов:

* прямым перебором по ArrayList, (array.contains())
* бинарным поиском по сортированному ArrayList, (Collections.binarySearch())
* поиском в HashSet, (setHash.contains())
* поиском в TreeSet. (setTree.contains())

1. Измерьте и сравните длительность каждого метода поиска. Формат вывода результатов поиска:

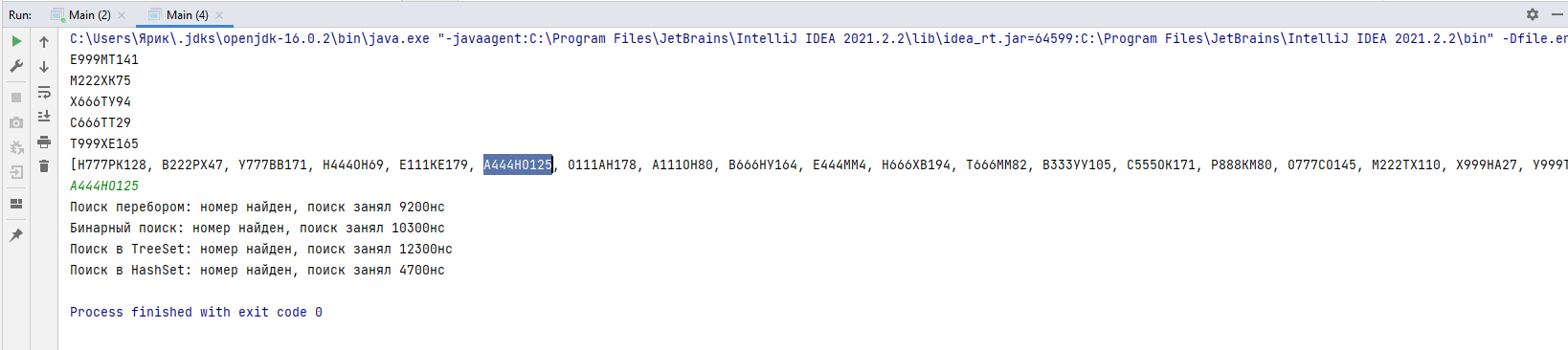
* Поиск перебором: номер найден/не найден, поиск занял 34нс
* Бинарный поиск: номер найден/не найден, поиск занял 34нс
* Поиск в HashSet: номер найден/не найден, поиск занял 34нс
* Поиск в TreeSet: номер найден/не найден, поиск занял 34нс

**Код программы**

**Класс Main**

**package** ru.mirea.Practice19\_20;  
  
**import** java.util.\*;  
  
**public class** Main {  
 **static** Map<Integer, String> *letters* = **new** HashMap<>();  
 **static int**[] *masDigits* = **new int**[]{0, 1, 2, 3, 4, 5, 6, 7, 8, 9};  
 **public static** String genNumber() {  
 **int** dig = (**int**) (Math.*random*() \* 10 + 0);  
 String ret = *letters*.get((**int**)(Math.*random*() \* 12 + 0)) + *masDigits*[dig]+*masDigits*[dig]+*masDigits*[dig] +  
 *letters*.get((**int**)(Math.*random*() \* 12 + 0))+*letters*.get((**int**)(Math.*random*() \* 12 + 0))+ (**int**)(Math.*random*() \* 199 + 1) ;  
 **return** ret;  
 }  
  
 **public static void** main(String[] args) {  
 *letters*.put(0, **"А"**);  
 *letters*.put(1, **"В"**);  
 *letters*.put(2, **"Е"**);  
 *letters*.put(3, **"К"**);  
 *letters*.put(4, **"М"**);  
 *letters*.put(5, **"Н"**);  
 *letters*.put(6, **"О"**);  
 *letters*.put(7, **"Р"**);  
 *letters*.put(8, **"С"**);  
 *letters*.put(9, **"Т"**);  
 *letters*.put(10, **"У"**);  
 *letters*.put(11, **"Х"**);  
 **for** (**int** i = 0; i < 5; i++) {  
 System.***out***.println(*genNumber*());  
 }  
 ArrayList<String> array = **new** ArrayList<>();  
 TreeSet<String> tree = **new** TreeSet<>();  
 HashSet<String> hash = **new** HashSet<>();  
  
 **for** (**int** i = 0; i < 20; i++){  
 String k = *genNumber*();  
 array.add(k);  
 tree.add(k);  
 hash.add(k);  
 }  
 System.***out***.println(array);  
  
 Scanner in = **new** Scanner(System.***in***);  
 String s = in.next();  
 **long** beginTime = System.*nanoTime*();  
 **boolean** b = array.contains(s);  
 **long** elapsedTime = System.*nanoTime*() - beginTime;  
 **if** (b) System.***out***.println(**"Поиск перебором: номер найден, поиск занял "** + elapsedTime + **"нс"**);  
 **else** System.***out***.println(**"Поиск перебором: номер не найден, поиск занял "** + elapsedTime +**"нс"**);  
   
 Collections.*sort*(array);  
 beginTime = System.*nanoTime*();  
 **int** cls;  
 cls = Collections.*binarySearch*(array, s);  
 elapsedTime = System.*nanoTime*() - beginTime;  
 **if** (cls >=0) System.***out***.println(**"Бинарный поиск: номер найден, поиск занял "** + elapsedTime + **"нс"**);  
 **else** System.***out***.println(**"Бинарный поиск: номер не найден, поиск занял "** + elapsedTime +**"нс"**);  
   
 beginTime = System.*nanoTime*();  
 b = tree.contains(s);  
 elapsedTime = System.*nanoTime*() - beginTime;  
 **if** (b) System.***out***.println(**"Поиск в TreeSet: номер найден, поиск занял "** + elapsedTime + **"нс"**);  
 **else** System.***out***.println(**"Поиск в TreeSet: номер не найден, поиск занял "** + elapsedTime +**"нс"**);  
   
 beginTime = System.*nanoTime*();  
 b = hash.contains(s);  
 elapsedTime = System.*nanoTime*() - beginTime;  
 **if** (b) System.***out***.println(**"Поиск в HashSet: номер найден, поиск занял "** + elapsedTime + **"нс"**);  
 **else** System.***out***.println(**"Поиск в HashSet: номер не найден, поиск занял "** + elapsedTime +**"нс"**);  
  
 }  
}

**Тестирование**



**Выводы**

По ходу работы был написан генератор автомобильных номеров, проведен поиск номера, введенного с консоли, а также мы измерили и сравнили длительность методов поиска: перебора, бинарного, поиска в HashSet и в TreeSet.

Git: <https://github.com/YaroslavChernov0220/PracticeMIREA>