|  |
| --- |
|  |
| 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 ПРАКТИЧЕСКОМУ ЗАНЯТИЮ №19-20**

Выполнил студент группы ИНБО-02-20 Судариков.Д.А.

Принял Степанов П.В.

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

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

Отметка о выполнении

**Москва – 2021 г.**

## **Задание**

Написать генератор «красивых» автомобильных номеров с помощью правил генерации номеров

Сгенерировать номера и провести поиск номера, введенного из консоли

Измерить и сравнить длительность каждого метода поиска.

## **Ход Работы**

В ходе выполнения работы были получены следующие исходные коды:

package ru.mirea.lab.lab19\_20;  
  
import javax.swing.plaf.synth.SynthLookAndFeel;  
import java.io.File;  
import java.io.FileNotFoundException;  
import java.io.PrintStream;  
import java.util.\*;  
  
public class NumberGenerator {  
 public static void main(String[] args){  
 int a=0,b=200;  
 int c = a + (int)(Math.*random*()\*b);  
 String[] letter = {"A","B","E","K","M","H","O","P","C","T","Y","X"};  
 Arrays.*sort*(letter);  
 ArrayList<String> arrayList = new ArrayList<>();  
 TreeSet<String> treeSet = new TreeSet<>();  
 HashSet<String> hashSet = new HashSet<>();  
  
 String num;  
 //одинаковые сочетания букв  
 for(String a123: letter) {  
 for(int reg = 1; reg <= 199; reg++) {  
 for(int j = 1; j <= 999; j++) {  
 num = a123 + j + j + j + a123 + a123 + reg;  
 arrayList.add(num);  
 treeSet.add(num);  
 hashSet.add(num);  
 }  
 }  
 }  
  
 for(int i = 111; i <= 999; i += 111) {  
 for (String a1 : letter) {  
 for (String a2 : letter) {  
 for (String a3 : letter) {  
 for (int reg = 1; reg <= 199; reg++) {  
 num =a1 + i+a2+a3+reg;  
 arrayList.add(num);  
 treeSet.add(num);  
 hashSet.add(num);  
 }  
 }  
 }  
 }  
 }  
 System.*out*.println(arrayList);  
 Scanner in = new Scanner(System.*in*);  
 boolean t;  
 long beginTime, elapsedTime;  
 int tt;  
 String k1 = in.next();  
 beginTime = System.*nanoTime*();  
 t = arrayList.contains(k1);  
 elapsedTime = System.*nanoTime*() - beginTime;  
 if (t) 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*(arrayList);  
 beginTime = System.*nanoTime*();  
 tt = Collections.*binarySearch*(arrayList, k1);  
 elapsedTime = System.*nanoTime*() - beginTime;  
 if (tt >=0) System.*out*.println("Поиск binarySearch: номер найден, поиск занял " + elapsedTime + "нс");  
 else System.*out*.println("Поиск binarySearch: номер не найден, поиск занял " + elapsedTime +"нс");  
 beginTime = System.*nanoTime*();  
 t = treeSet.contains(k1);  
 elapsedTime = System.*nanoTime*() - beginTime;  
 if (t) System.*out*.println("Поиск treeSet: номер найден, поиск занял " + elapsedTime + "нс");  
 else System.*out*.println("Поиск treeSet: номер не найден, поиск занял " + elapsedTime +"нс");  
 beginTime = System.*nanoTime*();  
 t = hashSet.contains(k1);  
 elapsedTime = System.*nanoTime*() - beginTime;  
 if (t) System.*out*.println("Поиск hashSet: номер найден, поиск занял " + elapsedTime + "нс");  
 else System.*out*.println("Поиск hashSet: номер не найден, поиск занял " + elapsedTime +"нс");  
  
 }  
}

## **Вывод**

В ходе выполнения данной практической работы я научился создавать генератор «красивых» автомобильных номеров с помощью правил генерации номеров, провёл поиск номера, введенного из консоли и научился измерять время работы методов на Java.