|  |
| --- |
|  |
| 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 Колмаков Е.Ю.

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 Generator**

package Lab19and20;  
  
public class Generator {  
 private char[] ch = new char[12];  
 private short reg;  
  
 public Generator() {  
 ch[0]='А';  
 ch[1]='В';  
 ch[2]='Е';  
 ch[3]='К';  
 ch[4]='М';  
 ch[5]='Н';  
 ch[6]='О';  
 ch[7]='Р';  
 ch[8]='С';  
 ch[9]='Т';  
 ch[10]='У';  
 ch[11]='Х';  
  
 }  
  
  
  
 public String num(){  
 reg = (short) ((short)(Math.*random*()\*198)+1);  
 String strReg="";  
 byte n = (byte) ((byte)(Math.*random*()\*9));  
 if (reg < 10)  
 {  
 strReg = "0";  
 }  
 strReg += Short.*toString*(reg);  
 byte i1 = (byte) ((byte)(Math.*random*()\*11));  
 byte i2 = (byte) ((byte)(Math.*random*()\*11));  
 byte i3 = (byte) ((byte)(Math.*random*()\*11));  
  
 return ch[i1]+Byte.*toString*(n)+Byte.*toString*(n)+Byte.*toString*(n)+ch[i2]+ch[i3]+strReg;  
 }  
}

**Класс Main**

package Lab19and20;  
  
import java.util.\*;  
  
public class Main {  
 public static void main(String[] args) {  
 int i = 0;  
 Generator g = new Generator();  
 ArrayList<String> nomera = new ArrayList<String>();  
 HashSet<String> nomeraForHash = new HashSet<String>();  
 TreeSet<String> nomeraForTree = new TreeSet<String>();  
 Scanner in = new Scanner(System.*in*);  
  
 while (i<2000236){  
 nomera.add(new String(g.num()));  
 nomeraForHash.add(new String(g.num()));  
 nomeraForTree.add(new String(g.num()));  
 i++;  
 }  
 System.*out*.println("Введите номер (используйте русский шрифт): ");  
 String nomer = in.next();  
 System.*out*.println("\n");  
 long start, finish;  
 //ArrayList, (array.contains())  
 System.*out*.println("Поиск прямым перебором по ArrayList, (array.contains())");  
 start = System.*nanoTime*();  
 boolean y = nomera.contains(nomer);  
 finish = System.*nanoTime*();  
 if (y)  
 System.*out*.println("Номер:"+ nomer+" найден. Время поиска: "+(finish - start) + "нс\n");  
 else  
 System.*out*.println("Номер:"+ nomer+" НЕ найден. Время поиска: "+(finish - start) + "нс\n");  
 //ArrayList, (Collections.binarySearch())  
 System.*out*.println("Поиск бинарным поиском по сортированному ArrayList, (Collections.binarySearch())");  
 Collections.*sort*(nomera);  
 start = System.*nanoTime*();  
 int resOfBinarySearch = Collections.*binarySearch*(nomera,nomer);  
 finish = System.*nanoTime*();  
 if (resOfBinarySearch>=0)  
 System.*out*.println("Номер:"+ nomer+" найден. Его позиция в базе: "+resOfBinarySearch+" Время поиска: "+(finish - start) + "нс\n");  
 else  
 System.*out*.println("Номер:"+ nomer+" НЕ найден. Время поиска: "+(finish - start) + "нс\n");  
 //HashSet, (setHash.contains())  
 System.*out*.println("Поиск в HashSet, (setHash.contains())");  
 start = System.*nanoTime*();  
 y = nomeraForHash.contains(nomer);  
 finish = System.*nanoTime*();  
 if (y)  
 System.*out*.println("Номер:"+ nomer+" найден. Время поиска: "+(finish - start) + "нс\n");  
 else  
 System.*out*.println("Номер:"+ nomer+" НЕ найден. Время поиска: "+(finish - start) + "нс\n");  
  
 //TreeSet. (setTree.contains())  
 System.*out*.println("Поиск в TreeSet. (setTree.contains())");  
 start = System.*nanoTime*();  
 y = nomeraForTree.contains(nomer);  
 finish = System.*nanoTime*();  
 if (y)  
 System.*out*.println("Номер:"+ nomer+" найден. Время поиска: "+(finish - start) + "нс\n");  
 else  
 System.*out*.println("Номер:"+ nomer+" НЕ найден. Время поиска: "+(finish - start) + "нс\n");  
  
 }  
}

**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/efim1111/KolmakovJavaINBO-02-20