**Практическая работа №11**

**Вариант №7 – Цепное хеширование.** Страховой полис: номер, компания, фамилия владельца.

**Автор: Николаев-Аксенов И. С.**

**Группа: ИКБО-20-19**

**Код программы:**

*Insurance.java:*

1. **public** **class** Insurance {
2. **int** number;
3. String company;
4. String surname;
6. **public** Insurance(**int** number, String company, String surname) {
7. **this**.number = number;
8. **this**.company = company;
9. **this**.surname = surname;
10. }
12. **public** **int** getNumber() {
13. **return** number;
14. }
16. @Override
17. **public** String toString() {
18. **return** "Insurance {" +
19. "number=" + number +
20. ", company='" + company + '**\'**' +
21. ", surname='" + surname + '**\'**' +
22. '}'+"**\n**";
23. }
24. }

*HashTable.java:*

1. **import** java.util.\*;
3. **public** **class** HashTable< E> {
4. ArrayList<LinkedList<E>>table;
5. **int** size;
7. **public** HashTable(**int** size) {
8. **this**.size = size;
9. **this**.table = **new** ArrayList<>(**this**.size);
11. **for**(**int** i=0;i < 10;i++){
12. table.add(**new** LinkedList<E>());
13. }
14. }
16. **int** hash(**int** value){
17. **return** (**int**)value%size;
18. }
20. **int** hash(E n) {
21. Insurance key = (Insurance) n;
22. **return** key.number % size;
23. }
25. **void** add(E b) {
26. table.get(hash(b)).addLast(b);
27. **if** (table.get(hash(b)).size() > 2) rehash();
28. }
30. **void** rehash(){
31. ArrayDeque<E> t =**new** ArrayDeque<>();
32. **for** (**int** i = 0; i < size; ++i) {
33. **for** (E el : table.get(i)) {
34. t.add(el);
35. }
36. }
37. size = size \* 2 + 1;
38. table.clear();
39. table = **new** ArrayList<>(size);
40. **for**(**int** i=0;i<size;i++){
41. table.add(**new** LinkedList<E>());
42. }
43. **while** (!t.isEmpty()) {
44. add(t.getFirst());
45. t.pop();
46. }
47. }
48. **void** search(**int** value){
49. **for**(E t:table.get(hash(value))){
50. Insurance c= (Insurance) t;
51. **if**(c.number == value){
52. System.out.println(c);
53. }
54. }
55. }
57. **void** delete(**int** value){
58. **for**(E t:table.get(hash(value))){
59. Insurance c= (Insurance) t;
60. **if**(c.number == value){
61. table.get(hash(value)).remove(t);
62. }
63. }
64. }
65. **void** print(){
66. **for**(**int** i=0;i<size;i++){
67. **if**(!table.get(i).isEmpty()) {
68. System.out.println(i+" : ");
69. **for** (**int** j = 0; j < table.get(i).size(); j++) {
70. System.out.println("**\t**"+table.get(i).get(j));
71. }
72. }
73. }
74. }
75. }

*StartInsurance.java:*

1. **public** **class** StartInsurance {
2. **public** **static** **void** main(String[] args) {
3. HashTable<Insurance> ht = **new** HashTable<>(10);
4. ht.add(**new** Insurance(123,"yandex","ivanov"));
5. ht.add(**new** Insurance(1005,"google","nikolaev"));
6. ht.add(**new** Insurance(5,"amazon","axenov"));
8. System.out.println("Вывод до рехеширования");
9. ht.print();
10. ht.add(**new** Insurance(2005,"netflix","ivanov"));
11. ht.add(**new** Insurance(3005,"tesla","nikolaev"));
13. System.out.println("Вывод после рехеширования");
14. ht.print();
15. System.out.println("**\n**");
17. ht.search(1005);
18. System.out.println("**\n**");
20. ht.delete(1005);
21. System.out.println("**\n**");
23. System.out.println("Вывод после удаления элемента");
24. ht.print();
25. }
26. }

**Результат выполнения программы:**



