**LAB NO: 14**

**NAME: NOOR FATIMA  
ROLL NO:21SW062**

**SECTION: II**

**TASK: 01**

The contact app on our phone contains a lot of contacts. In ContactApp(class) perform the following operations:

Display all contact

Search a contactby its name//name -> number

Add a new contact // name, number ,pos/index

Update the contact//name1, name2

Delete any contact //

**Solution:**

mport java.util.Scanner;public class LAB\_14 {  
 static private class ListNode {  
 Object key;  
 Object value;  
 ListNode next;  
 }  
  
 private ListNode[] table;  
 private int count;  
  
 public LAB\_14() {  
 table = new ListNode[64];  
 }  
  
 public LAB\_14(int initialSize) {  
 table = new ListNode[initialSize];  
 }  
  
 void dump() {  
 System.*out*.println();  
  
 for (int i = 0; i < table.length; i++) {  
 System.*out*.print(i + ":");  
 ListNode list = table[i]; // For traversing linked list number i.  
 while (list != null) {  
 System.*out*.print(" (" + list.key + "," + list.value + ")");  
 list = list.next;  
 }  
 System.*out*.println();  
 }  
 } // end dump()  
  
 public void put(Object key, Object value) {  
 int bucket = hash(key);  
 ListNode list = table[bucket]; // For traversing the linked list  
 while (list != null) {  
 if (list.key.equals(key))  
 break;  
 list = list.next;  
 }  
 if (list != null) {  
 list.value = value;  
 }  
 else {  
 if (count >= 0.75\*table.length) {  
 resize();  
 }  
 ListNode newNode = new ListNode();  
 newNode.key = key;  
 newNode.value = value;  
 newNode.next = table[bucket];  
 table[bucket] = newNode;  
 count++;  
 }  
 }  
  
 public Object get(Object key) {  
 int bucket = hash(key);  
 ListNode list = table[bucket];  
 while (list != null) {  
 if (list.key.equals(key))  
 return list.value;  
 list = list.next;  
 }  
 return null;  
 }  
  
 public void remove(Object key) {  
 int bucket = hash(key);  
 if (table[bucket] == null) {  
 return;  
 }  
 if (table[bucket].key.equals(key)) {  
 table[bucket] = table[bucket].next;  
 count--;  
 return;  
 }  
 ListNode prev = table[bucket];  
 ListNode curr = prev.next; // For traversing the list,  
 // starting from the second node.  
 while (curr != null && ! curr.key.equals(key)) {  
 curr = curr.next;  
 prev = curr;  
 }  
 if (curr != null) {  
 prev.next = curr.next;  
 count--;  
 }  
 }  
  
 public boolean containsKey(Object key) {  
 int bucket = hash(key);  
 ListNode list = table[bucket];  
 while (list != null) {  
 if (list.key.equals(key))  
 return true;  
 list = list.next;  
 }  
 return false;  
 }  
  
 public int size() {  
 return count;  
 }  
  
 private int hash(Object key) {  
 return (Math.*abs*(key.hashCode())) % table.length;  
 }  
  
 private void resize() {  
 ListNode[] newtable = new ListNode[table.length\*2];  
 for (int i = 0; i < table.length; i++) {  
 ListNode list = table[i];  
 while (list != null) {  
 ListNode next = list.next;  
 int hash = (Math.*abs*(list.key.hashCode())) % newtable.length;  
 list.next = newtable[hash];  
 newtable[hash] = list;  
 list = next;  
 }  
 }  
 table = newtable;  
 } // end resize()  
  
 } // end class

class TestHashTable {  
 public static void main(String[] args){  
  
 Scanner textIO=new Scanner(System.*in*);  
  
 LAB\_14 table = new LAB\_14(2);  
  
 String key,value;  
  
 while (true) {  
  
 System.*out*.println("\nMenu:");  
  
 System.*out*.println(" 1.Add contact: ");  
  
 System.*out*.println(" 2.Search Contact: ");  
  
 System.*out*.println(" 3.Update: ");  
  
 System.*out*.println(" 4.Delete Contact: ");  
  
 System.*out*.println(" 5.Display all contacts: ");  
  
 System.*out*.println(" 6. EXIT");  
  
 System.*out*.print("Enter your command: ");  
  
 switch (textIO.nextInt()) {  
  
 case 1:  
  
 System.*out*.print("\n Name = ");  
  
 key = textIO.next();  
  
 System.*out*.print("");  
  
 System.*out*.print(" Number = ");  
  
 value = textIO.next();  
  
 table.put(key,value);  
  
 System.*out*.print("");  
  
 break;  
  
 case 2:  
  
 System.*out*.print("\n Name = ");  
  
 key = textIO.next();  
  
 System.*out*.println(" Number is: " + table.get(key));  
  
 break;  
  
 case 3:  
  
 System.*out*.print("\n Name = ");  
  
 key = textIO.next();  
 if(table.containsKey(key)){  
 Object oldNumber=table.get(key);  
 System.*out*.println("Number is: "+oldNumber);  
 System.*out*.println("Enter new number to change: ");  
 Object number=textIO.next();  
 table.remove(key);  
// oldNumber=number;  
 table.put(key,number);  
  
 }  
  
 System.*out*.println(" containsKey(" + key + ") is "  
  
 + table.containsKey(key));  
  
 break;  
  
 case 4:  
  
 System.*out*.print("\n Name = ");  
  
 key = textIO.next();  
  
 table.remove(key);  
  
 break;  
  
 case 5:  
  
 table.dump();  
  
 break;  
  
 case 6:  
  
 return; // End program by returning from main()  
  
 default:  
  
 System.*out*.println(" Illegal command.");  
  
 break;  
  
 }  
  
 System.*out*.println("\nHash table size is " + table.size());  
 }}  
 }

**Output:**

"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.1\lib\idea\_rt.jar=60385:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.1\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\hp\IdeaProjects\DSA\_ALL\_LABS\out\production\DSA\_ALL\_LABS TestHashTable  
  
 Menu:  
 1.Add contact:  
 2.Search Contact:  
 3.Update:  
 4.Delete Contact:  
 5.Display all contacts:  
 6. EXIT  
 Enter your command: 1  
  
 Name = noor  
 Number = 12  
  
 Hash table size is 1  
  
 Menu:  
 1.Add contact:  
 2.Search Contact:  
 3.Update:  
 4.Delete Contact:  
 5.Display all contacts:  
 6. EXIT  
 Enter your command: 1  
  
 Name = fatima  
 Number = 1  
  
 Hash table size is 2  
  
 Menu:  
 1.Add contact:  
 2.Search Contact:  
 3.Update:  
 4.Delete Contact:  
 5.Display all contacts:  
 6. EXIT  
 Enter your command: 2  
  
 Name = fatima  
 Number is: 1  
  
 Hash table size is 2  
  
 Menu:  
 1.Add contact:  
 2.Search Contact:  
 3.Update:  
 4.Delete Contact:  
 5.Display all contacts:  
 6. EXIT  
 Enter your command: 3  
  
 Name = noor  
 Number is: 12  
 Enter new number to change:  
 14  
 containsKey(noor) is true  
  
 Hash table size is 2  
  
 Menu:  
 1.Add contact:  
 2.Search Contact:  
 3.Update:  
 4.Delete Contact:  
 5.Display all contacts:  
 6. EXIT  
 Enter your command: 5  
  
 0: (noor,14) (fatima,1)  
 1:  
  
 Hash table size is 2  
  
 Menu:  
 1.Add contact:  
 2.Search Contact:  
 3.Update:  
 4.Delete Contact:  
 5.Display all contacts:  
 6. EXIT  
 Enter your command: 6  
  
 Process finished with exit code 0