|  |  |  |
| --- | --- | --- |
| **Academic Year:** 2024-25 | **Year:** Third Year | **Term:** II |
| **PRN No.: 1012412079** | **Name: Ratnajeet Patil** | |
| **Subject:** DS2 | | |
| **Assignment No.**: 2 |  | |
| **Date:** |  | |

**Lab Assignment:**

**Title:** Write a program to create a Contact database of N clients. Make use of a hash table implementation to quickly look up a client's mobile number. (Apply Hash Function on mobile number) Write functions to 1) Add a new client 2) Display all the clients 3) Search information of specific client 4) Delete client information (logical deletion)

1. class Client {

2.     String name;

3.     long mobileNumber;

4.     String email;

5.     boolean deleted;

6.

7.     public Client(String name, long mobileNumber, String email) {

8.         this.name = name;

9.         this.mobileNumber = mobileNumber;

10.         this.email = email;

11.         this.deleted = false;

12.     }

13. }

14.

15. class HashTable {

16.     private Client[] table;

17.     private int size;

18.

19.     public HashTable(int size) {

20.         this.size = size;

21.         this.table = new Client[size];

22.     }

23.

24.     private int hashFunction(long mobileNumber) {

25.         return (int)(mobileNumber % size);

26.     }

27.

28.     public void addClient(String name, long mobileNumber, String email) {

29.         int index = hashFunction(mobileNumber);

30.         Client client = new Client(name, mobileNumber, email);

31.

32.         while (table[index] != null && !table[index].deleted) {

33.             index = (index + 1) % size;

34.         }

35.

36.         table[index] = client;

37.     }

38.

39.     public void displayClients() {

40.         System.out.println("Client Information:");

41.         for (Client client : table) {

42.             if (client != null && !client.deleted) {

43.                 System.out.println("Name: " + client.name + ", Mobile: " + client.mobileNumber + ", Email: " + client.email);

44.             }

45.         }

46.     }

47.

48.     public Client searchClient(long mobileNumber) {

49.         int index = hashFunction(mobileNumber);

50.

51.         while (table[index] != null) {

52.             Client client = table[index];

53.             if (client.mobileNumber == mobileNumber && !client.deleted) {

54.                 return client;

55.             }

56.             index = (index + 1) % size;

57.         }

58.         return null;

59.     }

60.

61.     public void deleteClient(long mobileNumber) {

62.         int index = hashFunction(mobileNumber);

63.

64.         while (table[index] != null) {

65.             Client client = table[index];

66.             if (client.mobileNumber == mobileNumber && !client.deleted) {

67.                 client.deleted = true;

68.                 System.out.println("Client " + client.name + " with Mobile " + mobileNumber + " deleted.");

69.                 return;

70.             }

71.             index = (index + 1) % size;

72.         }

73.         System.out.println("Client not found.");

74.     }

75. }

76.

77. public class ContactDatabase {

78.     public static void main(String[] args) {

79.         HashTable contactDb = new HashTable(10);

80.

81.         java.util.Scanner sc = new java.util.Scanner(System.in);

82.

83.         while (true) {

84.             System.out.println("\n1. Add a new client");

85.             System.out.println("2. Display all clients");

86.             System.out.println("3. Search for a client");

87.             System.out.println("4. Delete a client");

88.             System.out.println("5. Exit");

89.

90.             System.out.print("Enter your choice: ");

91.             int choice = sc.nextInt();

92.             sc.nextLine();

93.

94.             if (choice == 1) {

95.                 System.out.print("Enter client name: ");

96.                 String name = sc.nextLine();

97.                 System.out.print("Enter client mobile number: ");

98.                 long mobileNumber = sc.nextLong();

99.                 sc.nextLine();

100.                 System.out.print("Enter client email: ");

101.                 String email = sc.nextLine();

102.                 contactDb.addClient(name, mobileNumber, email);

103.             } else if (choice == 2) {

104.                 contactDb.displayClients();

105.             } else if (choice == 3) {

106.                 System.out.print("Enter mobile number to search: ");

107.                 long mobileNumber = sc.nextLong();

108.                 Client client = contactDb.searchClient(mobileNumber);

109.                 if (client != null) {

110.                     System.out.println("Name: " + client.name + ", Mobile: " + client.mobileNumber + ", Email: " + client.email);

111.                 } else {

112.                     System.out.println("Client not found.");

113.                 }

114.             } else if (choice == 4) {

115.                 System.out.print("Enter mobile number to delete: ");

116.                 long mobileNumber = sc.nextLong();

117.                 contactDb.deleteClient(mobileNumber);

118.             } else if (choice == 5) {

119.                 break;

120.             } else {

121.                 System.out.println("Invalid choice, please try again.");

122.             }

123.         }

124.

125.         sc.close();

126.     }

127. }

128.

129. class Client {

130.     String name;

131.     long mobileNumber;

132.     String email;

133.     boolean deleted;

134.

135.     public Client(String name, long mobileNumber, String email) {

136.         this.name = name;

137.         this.mobileNumber = mobileNumber;

138.         this.email = email;

139.         this.deleted = false;

140.     }

141. }

142.

143. class HashTable {

144.     private Client[] table;

145.     private int size;

146.

147.     public HashTable(int size) {

148.         this.size = size;

149.         this.table = new Client[size];

150.     }

151.

152.     private int hashFunction(long mobileNumber) {

153.         return (int)(mobileNumber % size);

154.     }

155.

156.     public void addClient(String name, long mobileNumber, String email) {

157.         int index = hashFunction(mobileNumber);

158.         Client client = new Client(name, mobileNumber, email);

159.

160.         while (table[index] != null && !table[index].deleted) {

161.             index = (index + 1) % size;

162.         }

163.

164.         table[index] = client;

165.     }

166.

167.     public void displayClients() {

168.         System.out.println("Client Information:");

169.         for (Client client : table) {

170.             if (client != null && !client.deleted) {

171.                 System.out.println("Name: " + client.name + ", Mobile: " + client.mobileNumber + ", Email: " + client.email);

172.             }

173.         }

174.     }

175.

176.     public Client searchClient(long mobileNumber) {

177.         int index = hashFunction(mobileNumber);

178.

179.         while (table[index] != null) {

180.             Client client = table[index];

181.             if (client.mobileNumber == mobileNumber && !client.deleted) {

182.                 return client;

183.             }

184.             index = (index + 1) % size;

185.         }

186.         return null;

187.     }

188.

189.     public void deleteClient(long mobileNumber) {

190.         int index = hashFunction(mobileNumber);

191.

192.         while (table[index] != null) {

193.             Client client = table[index];

194.             if (client.mobileNumber == mobileNumber && !client.deleted) {

195.                 client.deleted = true;

196.                 System.out.println("Client " + client.name + " with Mobile " + mobileNumber + " deleted.");

197.                 return;

198.             }

199.             index = (index + 1) % size;

200.         }

201.         System.out.println("Client not found.");

202.     }

203. }

204.

205. public class ContactDatabase {

206.     public static void main(String[] args) {

207.         HashTable contactDb = new HashTable(10);

208.

209.         java.util.Scanner sc = new java.util.Scanner(System.in);

210.

211.         while (true) {

212.             System.out.println("\n1. Add a new client");

213.             System.out.println("2. Display all clients");

214.             System.out.println("3. Search for a client");

215.             System.out.println("4. Delete a client");

216.             System.out.println("5. Exit");

217.

218.             System.out.print("Enter your choice: ");

219.             int choice = sc.nextInt();

220.             sc.nextLine();

221.

222.             if (choice == 1) {

223.                 System.out.print("Enter client name: ");

224.                 String name = sc.nextLine();

225.                 System.out.print("Enter client mobile number: ");

226.                 long mobileNumber = sc.nextLong();

227.                 sc.nextLine();

228.                 System.out.print("Enter client email: ");

229.                 String email = sc.nextLine();

230.                 contactDb.addClient(name, mobileNumber, email);

231.             } else if (choice == 2) {

232.                 contactDb.displayClients();

233.             } else if (choice == 3) {

234.                 System.out.print("Enter mobile number to search: ");

235.                 long mobileNumber = sc.nextLong();

236.                 Client client = contactDb.searchClient(mobileNumber);

237.                 if (client != null) {

238.                     System.out.println("Name: " + client.name + ", Mobile: " + client.mobileNumber + ", Email: " + client.email);

239.                 } else {

240.                     System.out.println("Client not found.");

241.                 }

242.             } else if (choice == 4) {

243.                 System.out.print("Enter mobile number to delete: ");

244.                 long mobileNumber = sc.nextLong();

245.                 contactDb.deleteClient(mobileNumber);

246.             } else if (choice == 5) {

247.                 break;

248.             } else {

249.                 System.out.println("Invalid choice, please try again.");

250.             }

251.         }

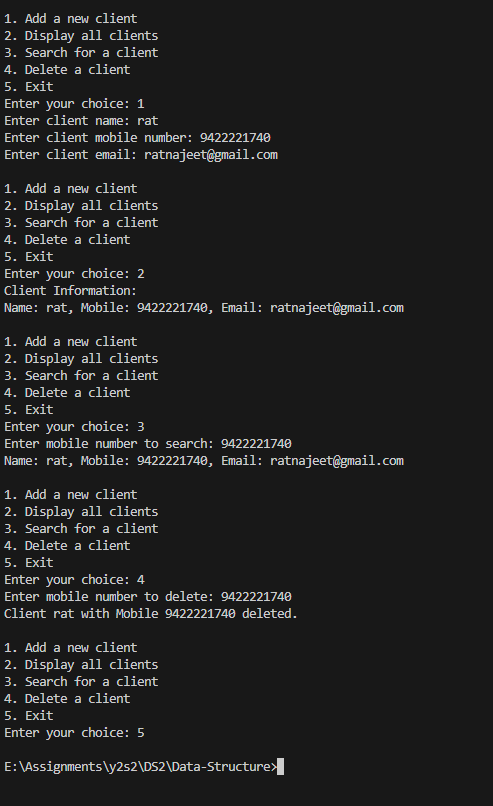
252.

253.         sc.close();

254.     }

255. }

256.

****