Phonebook Application Java Code

```
import java.util.Scanner;

class Contact {
    String name;
    String phoneNumber;

    Contact(String name, String phoneNumber) {
        this.name = name;
        this.phoneNumber = phoneNumber;
    }
}

public class Phonebook {
    private static final int MAX_CONTACTS = 100;
    private Contact[] contactList = new Contact[MAX_CONTACTS];
    private int currentSize = 0;

    public void insertContact(String name, String phoneNumber) {
        if (currentSize < MAX_CONTACTS) {
            contactList[currentSize++] = new Contact(name, phoneNumber);
            System.out.println("Contact added: " + name);
        } else {
            System.out.println("Phonebook is full!");
        }
    }
```

```
public Contact searchContact(String name) {
   for (int i = 0; i < currentSize; i++) {
      if (contactList[i].name.equalsIgnoreCase(name)) {
         return contactList[i];
      }
   }
   return null;
}

public void displayAllContacts() {
   if (currentSize == 0) {
      System.out.println("No contacts available.");
   } else {
      for (int i = 0; i < currentSize; i++) {
         System.out.println(contactList[i].name + ": " + contactList[i].phoneNumber);
      }
   }
}

public boolean deleteContact(String name) {
   for (int i = 0; i < currentSize; i++) {
      if (contactList[i].name.equalsIgnoreCase(name)) {
         contactList[i] = contactList[--currentSize]; // Replace with last contact
         System.out.println("Contact deleted: " + name);
         return true;
      }
   }
```

```
    System.out.println("Contact not found: " + name);
    return false;
}

public boolean updateContact(String name, String newPhoneNumber) {
    for (int i = 0; i < currentSize; i++) {
        if (contactList[i].name.equalsIgnoreCase(name)) {
            contactList[i].phoneNumber = newPhoneNumber;
            System.out.println("Contact updated: " + name);
            return true;
        }
    }
    System.out.println("Contact not found: " + name);
    return false;
}

public void sortContacts() {
    for (int i = 0; i < currentSize - 1; i++) {
        for (int j = 0; j < currentSize - i - 1; j++) {
            if (contactList[j].name.compareTo(contactList[j + 1].name) > 0) {
                Contact temp = contactList[j];
                contactList[j] = contactList[j + 1];
                contactList[j + 1] = temp;
            }
        }
    }
    System.out.println("Contacts sorted.");
}
```

```
public void analyzeSearchEfficiency(String targetName) {
    long startTime = System.nanoTime();
    searchContact(targetName);
    long endTime = System.nanoTime();
    System.out.println("Search time: " + (endTime - startTime) + " nanoseconds");
}

public static void main(String[] args) {
    Phonebook phonebook = new Phonebook();
    Scanner scanner = new Scanner(System.in);
    int choice;

    do {
        System.out.println("\nPhonebook Menu:");
        System.out.println("1. Insert Contact");
        System.out.println("2. Search Contact");
        System.out.println("3. Display All Contacts");
        System.out.println("4. Delete Contact");
        System.out.println("5. Update Contact");
        System.out.println("6. Sort Contacts");
        System.out.println("7. Analyze Search Efficiency");
        System.out.println("0. Exit");
        System.out.print("Enter your choice: ");
        choice = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        switch (choice) {
```

```
            case 1:
                System.out.print("Enter name: ");
                String nameToInsert = scanner.nextLine();
                System.out.print("Enter phone number: ");
                String phoneToInsert = scanner.nextLine();
                phonebook.insertContact(nameToInsert, phoneToInsert);
                break;
            case 2:
                System.out.print("Enter name to search: ");
                String nameToSearch = scanner.nextLine();
                Contact foundContact = phonebook.searchContact(nameToSearch);
                if (foundContact != null) {
                    System.out.println("Found: " + foundContact.name + " - " +
foundContact.phoneNumber);
                } else {
                    System.out.println("Contact not found!");
                }
                break;
            case 3:
                phonebook.displayAllContacts();
                break;
            case 4:
                System.out.print("Enter name to delete: ");
                String nameToDelete = scanner.nextLine();
                phonebook.deleteContact(nameToDelete);
                break;
            case 5:
                System.out.print("Enter name to update: ");
```

```
                String nameToUpdate = scanner.nextLine();
                System.out.print("Enter new phone number: ");
                String newPhone = scanner.nextLine();
                phonebook.updateContact(nameToUpdate, newPhone);
                break;
            case 6:
                phonebook.sortContacts();
                break;
            case 7:
                System.out.print("Enter name to analyze search efficiency: ");
                String nameToAnalyze = scanner.nextLine();
                phonebook.analyzeSearchEfficiency(nameToAnalyze);
                break;
            case 0:
                System.out.println("Exiting...");
                break;
            default:
                System.out.println("Invalid choice! Please try again.");
        }
    } while (choice != 0);

    scanner.close();
  }
}
```