```
CLASS Contact
  PRIVATE STRING name
  PRIVATE STRING number

  Contact(name, number)
    SET this.name = name
    SET this.number = number

  getName() RETURNS STRING
    RETURN name

  getNumber() RETURNS STRING
    RETURN number

  setName(name)
    SET this.name = name

  setNumber(number)
    SET this.number = number

  compareTo(other: Contact) RETURNS INTEGER
    RETURN this.name.compareToIgnoreCase(other.name)

  toString() RETURNS STRING
    RETURN name + ": " + number

END CLASS
```

```
CLASS ContactManager
  PRIVATE LIST<Contact> contacts

  ContactManager()
    INITIALIZE contacts as empty list

  addContact(name, number)
    APPEND new Contact(name, number) to contacts

  searchContacts(query) RETURNS LIST<Contact>
    INITIALIZE results as empty list
    FOR EACH contact IN contacts DO
      IF contact.getName().toLowerCase().contains(query.toLowerCase()) OR
        contact.getNumber().contains(query) THEN
         APPEND contact to results
      END IF
    END FOR
    RETURN results

  deleteContact(index: INTEGER)
    IF index >= 0 AND index < SIZE(contacts) THEN
      REMOVE contact at index from contacts
    END IF

  updateContact(index: INTEGER, name, number)
    IF index >= 0 AND index < SIZE(contacts) THEN
      SET contact = contacts[index]
      contact.setName(name)
      contact.setNumber(number)
    END IF

  sortContacts()
    SORT contacts using compareTo method

  getAllContacts() RETURNS LIST<Contact>
    RETURN COPY OF contacts

END CLASS

CLASS PhonebookApp
  PRIVATE ContactManager contactManager
  PRIVATE TEXT_FIELD searchField, nameField, numberField
  PRIVATE LIST<Contact> contactList
  PRIVATE LIST_MODEL<Contact> listModel

  PhonebookApp()
    SET contactManager = new ContactManager()
    SET window title to "Phonebook Application"
    SET window size to (500, 500)
    SET close operation to EXIT
    SET layout to BorderLayout with padding

    // Initialize main panel
    INITIALIZE mainPanel with padding

    // Create Search Panel
```

```
INITIALIZE searchPanel
SET searchField to input field of length 20
SET searchButton to button labeled "Search"
ADD label, searchField, and searchButton to searchPanel

// Create Add Contact Panel
INITIALIZE addPanel with grid layout
INITIALIZE nameField and numberField with length 15
SET addButton to button labeled "Add Contact"

// Add components to addPanel with grid constraints

// Create Contact List
INITIALIZE listModel
SET contactList to display list using listModel
INITIALIZE scrollPane with contactList

// Create Buttons Panel
INITIALIZE buttonPanel with vertical layout
SET deleteButton, updateButton, and sortButton
ADD buttons to buttonPanel

// Arrange panels in mainPanel
ADD searchPanel to mainPanel
ADD addPanel and buttonPanel to west side of mainPanel
ADD scrollPane to mainPanel

// Setup Event Listeners
```

```
  ADD event handler for searchButton calling searchContacts()
  ADD event handler for addButton calling addContact()
  ADD event handler for deleteButton calling deleteContact()
  ADD event handler for updateButton calling updateContact()
  ADD event handler for sortButton calling sortContacts()

  CALL refreshContactList()

searchContacts()
  SET query to value of searchField
  CLEAR listModel
  FOR EACH contact IN contactManager.searchContacts(query) DO
    ADD contact to listModel
  END FOR

addContact()
  SET name to value of nameField
  SET number to value of numberField
  IF name AND number are not empty THEN
    CALL contactManager.addContact(name, number)
    CLEAR nameField and numberField
    CALL refreshContactList()
  ELSE
    SHOW error message for missing input
  END IF

deleteContact()
  SET selectedIndex to index of selected contact in contactList
  IF selectedIndex is valid THEN
    CONFIRM deletion from user
    IF user confirms THEN
      CALL contactManager.deleteContact(selectedIndex)
      CALL refreshContactList()
    END IF
  ELSE
    SHOW error message for no selection
  END IF

updateContact()
  SET selectedIndex to index of selected contact in contactList
  IF selectedIndex is valid THEN
    GET selected contact from listModel
    INITIALIZE nameField and numberField with selected contact data
    SHOW dialog for updating contact
    IF user confirms THEN
      SET name and number to values of input fields
      IF name AND number are not empty THEN
        CALL contactManager.updateContact(selectedIndex, name, number)
        CALL refreshContactList()
      ELSE
        SHOW error message for empty fields
      END IF
    END IF
  ELSE
    SHOW error message for no selection
  END IF
  sortContacts()
    CALL contactManager.sortContacts()
    CALL refreshContactList()

  refreshContactList()
    CLEAR listModel
    FOR EACH contact IN contactManager.getAllContacts() DO
      ADD contact to listModel
    END FOR

  main()
    CALL to set look and feel of UI
    CREATE new PhonebookApp instance
    SET window to be visible

END CLASS
```