

Address Book

Introduction

Your parents currently have all names and addresses of their relatives, friends, and contacts stored in a text file in the following format:

Name
Street Address
City, State Zip
Country
Phone Number

Your parents have been impressed with your programs and need your help to automate their address book.

Problem Statement

Develop an **AddressBook.java** class to automate an address book.

Design Requirements

Your implementation must meet the following design requirements.

- The program must use the **LList.java** class developed in class to organize your parents' address book.
- The user of your program should be able to perform the following operations on their address book. As a result, your program should have methods for these operations. If the user chooses to view, delete, or modify an existing address entry; they will use the last name field to identify the address record.
 1. Add a new address record
 2. View an existing address record
 3. Delete an existing address record
 4. Modify an existing address record
 5. Save the entire address book in a file
 6. Retrieve all address entries using zip code
 7. Print the entire address book on the screen
- The program should display a menu allowing the user to select one of the operations listed above. Provide an additional option to allow the user to quit your program.
- According to user's selection, your program calls the correspondent method to perform the operation. Your program exits only when the user chooses to do so.

Program Input

The program gets its input from a data file named **addresses.txt** with data in an address record is organized as described above. Note that there is a blank line between two consecutive address records.

Program Output

The output of the program depends on the user's choice.

Submission Instructions

When you are satisfied with your implementation, and after testing it thoroughly, submit the **AddressBook.java** file via Blackboard Learn.