

```

--
-- A simple address book application
--
-- Author: Gabriele Keller
--
module Main where

import IO
import AddressDB
import ADB_UserInterface

-- read address data from user and
-- insert new entry into address book
doInsert :: AddressBook -> IO ()
doInsert db =
  do
    adr <- readAddressDialogue
    mainLoop (insertAddress db adr)

-- read search string from user, search
-- address book and display matches
doSearch :: AddressBook -> IO ()
doSearch db =
  do
    searchStr <- readSearchStr
    let adrList = findMatches db searchStr
    displayMatches adrList
    mainLoop db

-- read search string from user, search
-- address book, display matches. Let user
-- choose which of the matches to delete,
-- update address book accordingly
doDelete :: AddressBook -> IO ()
doDelete db =
  do
    searchStr <- readSearchStr
    let adrList = findMatches db searchStr
    displayMatches adrList
    let len = length adrList
    ind <- readChoice (len - 1 )
    mainLoop (deleteAddress db (adrList!!ind))

doChange :: AddressBook -> IO ()
doChange db =
  do
    searchStr <- readSearchStr
    let adrList = findMatches db searchStr
    displayMatches adrList
    let len = length adrList
    ind <- readChoice (len - 1 )
    let adr = (adrList!!ind)
    newAddr <- changeAddressDialogue adr
    mainLoop (changeAddress db adr newAddr)

-- main
mainLoop :: AddressBook -> IO ()
mainLoop db =
  do
    cmd <- mainDialog
    case cmd of
      AddEntry -> doInsert db
      Search   -> doSearch db
      Delete   -> doDelete db
      Change   -> doChange db
      Quit     -> return ()

```

```
startDB:: IO ()
startDB = mainLoop newAddressBook

main:: IO ()
main = do
    hSetBuffering stdout NoBuffering
    startDB
```