Table of Contents

[Project Overview 2](#__RefHeading___Toc4258_1370385848)

[Top Level Functions 3](#__RefHeading___Toc4260_1370385848)

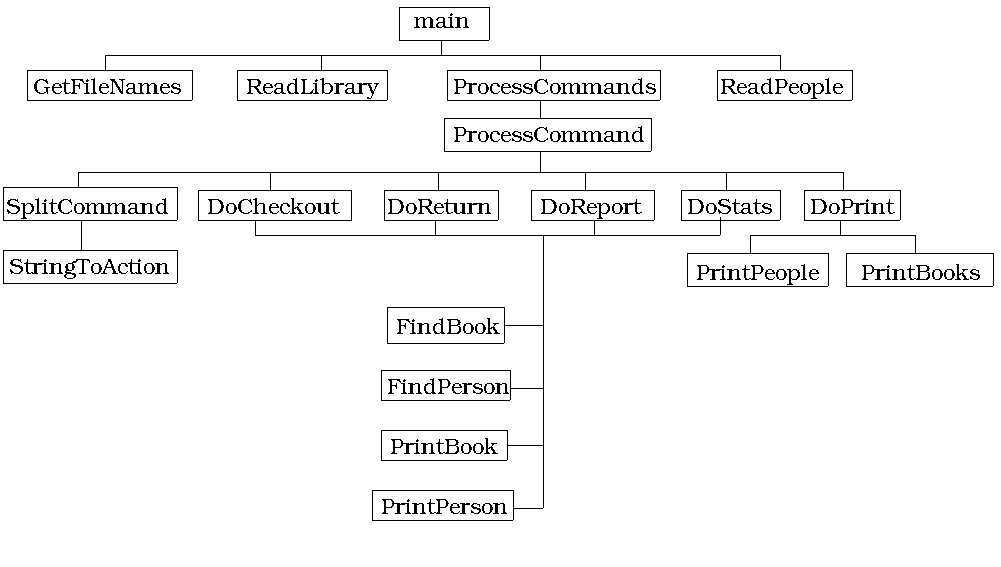
[ActionT (Enumeration) 8](#__RefHeading___Toc2147_279954108)

[BookT (Structure) 9](#__RefHeading___Toc4283_1370385848)

# Project Overview

This program simulates a simple library. There is a basic database that contains information about books and another containing information about readers. The program allows readers to check out and return books. In addition, simple statistics on the books and readers are collected.

**Structure Chart**



# Top Level Functions

**Main**

Narrative: This is an extremely high level function which just calls the component parts.

The program employed two arrays. Both are static and of fixed size. One represents the books in the library and the other the readers. These are called the library databases.

Get the file names

Read in the library

Read in the people

Process the commands.

**Function:** GetFileNames

Narrative: Get the master file name from the user, and return the names for each of the database components.

Input: None

Output: Three strings, the names of each of the database files and the command files.

While the user does not provide a valid file name.

Prompt the user for a file name.

Read the database and command file names from the user supplied file.

Return these names.

**Function:** ReadLibrary

Narrative: This function reads in the books stored in the book database. Note this function may produce errors if the book file is unavailable or the book file contains too many books.

Input: The name of the book database file.

Output: An array of books and the count of books in the library.

Open the book file

If unable to open the file

report an error

Otherwise

for each line in the file

Read the book and add it to the library

If there is no room

Report an error.

**Function:** ReadPeople

Narrative: This function reads in the people stored in the reader database. Note this function may produce errors if the reader file is unavailable or the reader file contains too many readers.

Input: The name of the reader database file.

Output: An array of books and the count of readers in the library.

Open the reader file

If unable to open the file

report an error

Otherwise

for each line in the file

Read the reader and add it to the library

If there is no room

Report an error.

**Function:** ProcessCommands

Narrative: This function will read and process the commands from the command file. It will produce an error if the file is not available.

Input: Command file name, library and person database.

Output: None

Open the command file

while there are commands

process the command

**Function:** ProcessCommand

Narrative: This function is responsible for processing an individual command.

Input: The command, and the library and person databases

Output: none.

Convert the command to an action.

Based on the action call the appropriate command processing routine.

**Function:** DoCheckout

Narrative: This routine will loan a book to a user if the user does not have a book, the book is in the library and the book is not checked out.

Input: The user, book and library databases.

Output: The updated library databases

If the reader does not exist (FindReader)

print an error

If the book does not exist (FindBook)

print an error

else

if the reader already has a book

print error

if the book is alreadychecked out

print an error

else

loan the book to the user

**Function:** DoReturn

Narrative: This routine attempts to return a book. It will produce errors if the book is not checked out or if the book does not exist.

Input: The book title and the library databases

Output: The modified databases

Find the book (FindBook)

If it is not present

print an error

If it is not checked out

print an error

else

remove the book from the user

return the book

update statistics.

**Function:** DoReport

Narrative: This routine will print information about a reader.

Input: The reader’s name and the library databases

Output: none

Find the reader

If they exist

PrintPerson

else

print an error message.

**Function:** DoStats

Narrative: This routine will print the stats for a book.

Input: The book title and the library databases

Output: none

Find the book

If the book exists

PrintBook

else

print an error

**Function:** DoPrint

Narrative: Prints either all of the books or all of the readers.

Input: The command along with the library databases.

Output:

If cmd is Books

PrintBooks

else

PrintPeople

**Function:** FindPerson

Narrative: This is a linear search of the person database

Input: The person name and the book array

Output: The position of the person, NOT\_FOUND

Linear search.

**Function:** FindBook

Narrative: This is a linear search of the book database

Input: The book array and the book’s name.

Output: The position of the book or NOT\_FOUND

Linear search.

**Function:** PrintPerson

Narrative: Prints a formatted report describing a person

Input: The record for a person

Output: None

**Function:** PrintBook

Narrative: Print a formatted report describing a book.

Input: The record for a book

Output: None

**Function:** PrintBooks

Narrative: Prints all the books.

Input: The book array

Output: none

For each book

PrintBook

**Function:** PrintPeople

Narrative: Prints all of the people.

Input: The people array

Output: none

For each person

PrintPerson

# ActionT (Enumeration)

This enumeration provides a list of all possible command actions.

Properties

Domain

CHECKOUT, REPORT, RETURN, STATS, PRINT, UNKNONW

Behaviors

One function that takes a string an returns the enumeration type.

Discussion:

none.

**Function:** StringToActionT

Narrative: This takes a string and returns the appropriate actionT.

Input: A string

Output: The corresponding action.

# BookT (Structure)

This structure represents a book.

Properties

Title: a string

Author: a string

Page Count: an int

borrower: the index into the person array of the person holding the book.

ReaderCount: the number of readers who have read the book.

Present: a bool indicating if the book is present in the library.

Behaviors

none

Discussion:

none

PersonT (Structure)

This structure holds a person.

Properties

Name: a string

booksRead an int describing how many books the person has read.

PagesRead: the total number of pages read by this individual.

Book: an index into the book array

HasBook: a bool, true if the person has a book.

Behaviors

None

Discussion:

None