Class: BookHead:

Parameters
BookLinkedList next -> First node in the linked list.

Methods:
Void addBook(String isbn, String title, String author, into totalCopies)-> creates a new Book and adds it to the end of the linked list.
Void addBook(Book book) -> Takes an existing book object and adds it to the end of the linked list
Int removeBook(String ISBN)-> searches for a book and removes it from the LL. Returns a 0 on success and a –1 on failure.
BookHead findBookByString(String bookString) takes a string and checks if it matches ISBN, Title, or Author of all books. Returns a Linked List head of all matching books.
Void printList() prints the book info of all nodes in the linked list.

Class: SubscriberHead

Parameters
SubscriberLinkedList next -> first node in the linked list.

Methods
Void addSubscriber(String id, String fullName, String userType) -> creates a new Subscriber and adds it to the end of the linked list.
Void addSubscriber(Subscriber subscriber) -> adds an existing subscriber to the end of the linked list.
Int removeSubscriber(String id) -> removes the subscriber with the matching id. Returns a 0 on success and a –1 on failure
SubscriberHead findSubscriberByString(String subscriberString) -> returns a SubscriberHead containing the subscriber with the id that matches the subscriberString.
Void printList() -> Prints all the subscribers in the LinkedList.

Class: BookLinkedList

Parameters:
BookLinkedList next -> next node in the LL
Book Data -> Book obj in the LL

Methods:
//Constructor//
//getters and setters//

Class: SubscriberLinkedList

Parameters
SubscriberLinkedList next
Subscriber data -> the subscriber object held by the LL

Methods
//constructor//
//getters & setters//

Class: Book

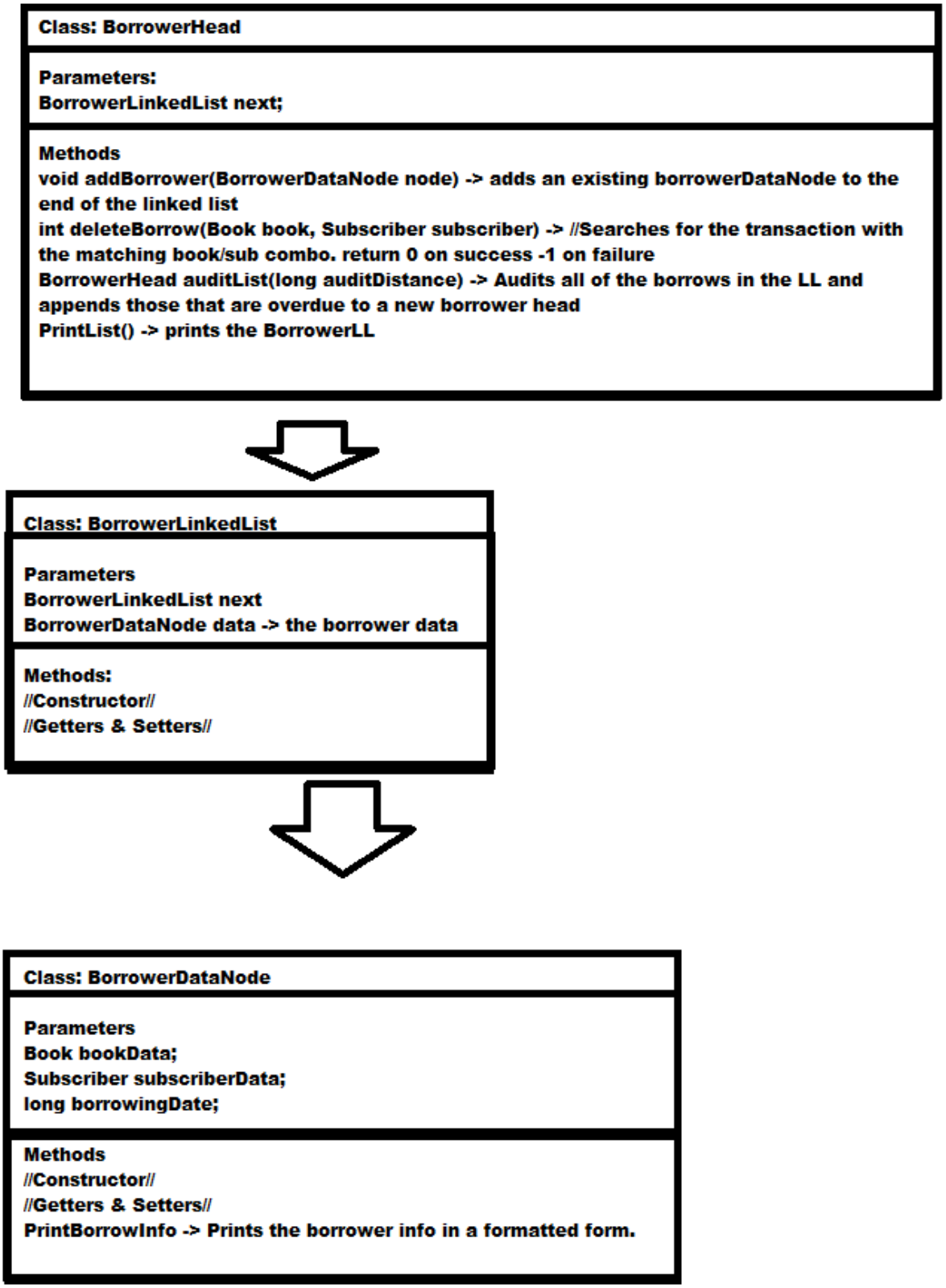
Parameters:
String isbn -> Unique identifier
String title -> Book Title
String author -> author of book
BorrowerHead borrowerLL-> A linked list of the current borrowers of the book
int totalCopies -> total copies of book in library
int availableCopies -> number of unchecked out copies.
int timesCheckedOut -> number of times book is checked out

Methods:
Constructor -> creates Book object.
getters/setters -> get/set parameter
Void PrintSubscribers -> prints all the subscribers in the borrowerLL.

Class: Subscriber

Parameters
String id -> unique ID of the customer
String fullName -> name of user
String userType -> Student, professor, etc.
BorrowerLL borrowerLL -> Linked list of borrowed books

Methods
//constructor//
//getters & setters//
Int deleteBorrow(Book book) -> removes a borrow from its borrowerLL.
Int CurrentlyCheckedOut(Book book) -> Gets the number of copies the user has of book.



Class: BookHead:

- Parameters
    - BookLinkedList next -> First node in the linked list.
- Methods:

- Void addBook(String isbn, String title, String author, into totalCopies)-> creates a new Book and adds it to the end of the linked list.
- Void addBook(Book book) -> Takes an existing book object and adds it to the end of the linked list
- Int removeBook(String ISBN)-> searches for a book and removes it from the LL. Returns a 0 on success and a –1 on failure.
- BookHead findBookByString(String bookString) takes a string and checks if it matches ISBN, Title, or Author of all books. Returns a Linked List head of all matching books.
- Void printList() prints the book info of all nodes in the linked list.

Class: BookLinkedList

- Parameters:
  - BookLinkedList next -> next node in the LL
  - Book Data -> Book obj in the LL
- Methods:
  - //Constructor//
  - //getters and setters//

Class: Book

- Parameters:
  - String isbn -> Unique identifier
  - String title -> Book Title
  - String author -> author of book
  - BorrowerHead borrowerLL-> A linked list of the current borrowers of the book
  - int totalCopies -> total copies of book in library
  - int availableCopies -> number of unchecked out copies.
  - int timesCheckedOut -> number of times book is checked out
- Methods:
  - Constructor -> creates Book object.
  - getters/setters -> get/set parameter
  - Void PrintSubscribers -> prints all the subscribers in the borrowerLL.

Class: SubscriberHead

- Parameters
  - SubscriberLinkedList next -> first node in the linked list.
- Methods
  - Void addSubscriber(String id, String fullName, String userType) -> creates a new Subscriber and adds it to the end of the linked list.
  - Void addSubscriber(Subscriber subscriber) -> adds an existing subscriber to the end of the linked list.

- Int removeSubscriber(String id) -> removes the subscriber with the matching id. Returns a 0 on success and a –1 on failure
- SubscriberHead findSubscriberByString(String subscriberString) -> returns a SubscriberHead containing the subscriber with the id that matches the subscriberString.
- Void printList() -> Prints all the subscribers in the LinkedList.

Class: SubscriberLinkedList

- Parameters
  - SubscriberLinkedList next
  - Subscriber data -> the subscriber object held by the LL
- Methods
  - //constructor//
  - //getters & setters//

Class: Subscriber

- Parameters
  - String id -> unique ID of the customer
  - String fullName -> name of user
  - String userType -> Student, professor, etc.
  - BorrowerLL borrowerLL -> Linked list of borrowed books
- Methods
  - //constructor//
  - //getters & setters//
  - Int deleteBorrow(Book book) -> removes a borrow from its borrowerLL.
  - Int CurrentlyCheckedOut(Book book) -> Gets the number of copies the user has of book.

Class: BorrowerHead

- Parameters:
  - BorrowerLinkedList next;
- Methods
  - void addBorrower(BorrowerDataNode node) -> adds an existing borrowerDataNode to the end of the linked list
  - int deleteBorrow(Book book, Subscriber subscriber) -> //Searches for the transaction with the matching book/sub combo. return 0 on success -1 on failure
  - BorrowerHead auditList(long auditDistance) -> Audits all of the borrows in the LL and appends those that are overdue to a new borrower head
  - PrintList() -> prints the BorrowerLL

Class: BorrowerLinkedList

- Parameters

    - BorrowerLinkedList next
    - BorrowerDataNode data -> the borrower data
- Methods:
    - //Constructor//
    - //Getters & Setters//

Class: BorrowerDataNode

- Parameters
    - Book bookData;
    - Subscriber subscriberData;
    - long borrowingDate;
- Methods
    - //Constructor//
    - //Getters & Setters//
    - PrintBorrowInfo -> Prints the borrower info in a formatted form.

Class: Main

Functions: The below categories are just to group functions by their function, they will not be combined in the code itself.

- Menus:
    - BookLinkedList MenuGetBook(Scanner, BookHead) -> shows the prompts required to find a book
    - SubscriberLinkedList MenuGetSubscriber (Scanner, SubscriberHead) -> shows the prompts required to find a subscriber
    - Void MenuCheckout(Scanner, BookHead, SubscriberHead) -> shows the prompts required for checking out a book. Calls menuGetBook and menuGetSubscriber
    - MenuBookOperations(Scanner, BookHead, SubscriberHead) -> shows the prompts necessary for performing book operations. Including: adding, updating, deleting, finding, auditing, and printing books.
    - MenuSubscriberOperations (Scanner, SubscriberHead)-> shows the prompts necessary for performing subscriber operations. Including: Adding, updating, deleting, printing, and finding subscribers.
    - MenuAuditOperations(Scanner, SubscriberHead) -> performs an audit of all the users
- Linked list creation:
    - BookLL ReadBooks (String filename) -> creates a linked list of Book objects and returns the head of the list. String should be “Books.txt”.
    - SubscriberLL ReadSubscribers(String filename) -> creates a linked list of subscribers, returns head of SubscriberLL. Filename should be“Subscribers.txt”.
    - void readBorrowerInfo((String filename, BookHead, SubscriberHead) -> Populates the Book and Subscriber’s BorrowersLL. filename should be of “BorrowInfo.txt”
- File maintenance

    - void writeBooks(String fileName , BookHead) -> Writes to “Books.txt” to show additions and deletions to the library. Should be called after Book additions and deletions
    - void writeSubscribers(String fileName , SubscriberHead) -> Writes to “Subscribers.txt” to show additions and deletions of users. Should be called after Subscriber additions and deletions
    - void writeBorrowerInfo(String fileName , SubscriberHead) -> Writes to “BorrowInfo.txt” to show changes in book borrowing. Book checkouts are grouped by subscriber, so the Subscriber IDs should be all in one place. Should be called after every checkout / check in.
- Management
    - int checkoutBook(Subscriber subscriber, Book book) -> Checks to see if user has already checked out 2 copies of the book, if not, a new BorrowerLinkedList node is created and added to the Book’s and Subscriber’s BorrowerLinkedList. The Book.availableCopies is then reduced by one. Returns 0 on success, returns -1 on failure (user has already checked out maxBookCopies of this book or the book has 0 available copies.)
    - void checkinBook(Subscriber subscriber, Book, book) -> removes the first BorrowerLinkedList node that contains the Subscriber and the Book from both the Book’s and Subscriber’s BorrowerLinkedList. Increases the book’s availabile copies.
- Statistics
    - Void ShowMostBorrowedBook(BookLL); - shows the most borrowed books.

File structures:

- Books.txt
    - ISBN
    - Title
    - Author
    - TotalCopies
    - AvailableCopies
    - TimesCheckedOut
    - …
- Subscribers.txt
    - ID
    - FullName
- BorrowerInfo.txt
    - ID
    - ISBN
    - Date