

04 – Case Study – A Library Application

OBJECT-ORIENTED PROGRAMMING

SPECIFICATION

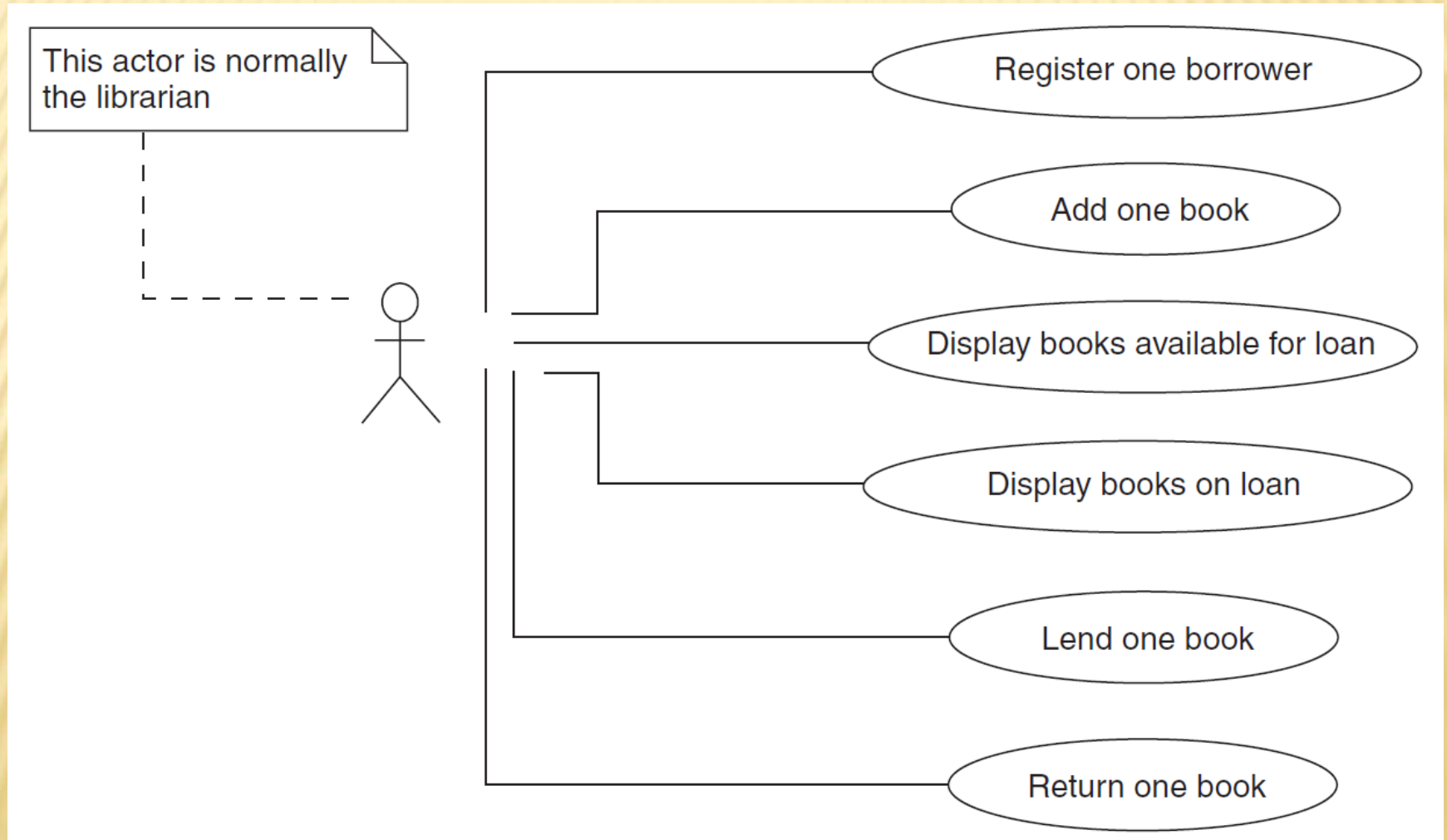
✕ The library:

- + name
- + books:
 - ✕ title,
 - ✕ author
 - ✕ unique catalogue number.
- + borrowers:
 - ✕ name
 - ✕ borrow a book
 - ✕ return a book
- + register a new borrower
- + display those books:
 - ✕ available for loan
 - ✕ already out on loan.

USE-CASES

- ✗ Register a new borrower
- ✗ Add a new book to the stock
- ✗ Produce a list of books available for loan
- ✗ Produce a list of books out on loan
- ✗ Record that a book has been borrowed and
- ✗ Record that a book has been returned

USE-CASE DIAGRAM



TEST-CASES

1. Register one borrower

A borrower is successfully registered with the library.

2. Add one book

A book is successfully added to the library stock and is available for loan.

3. Display books available for loan

A display of each book available for loan when there are zero or more books in the library.

TEST-CASES (CONT.)

4. Display books out on loan

A display of each book out on loan when there are zero or more books in the library.

5. Lend one book

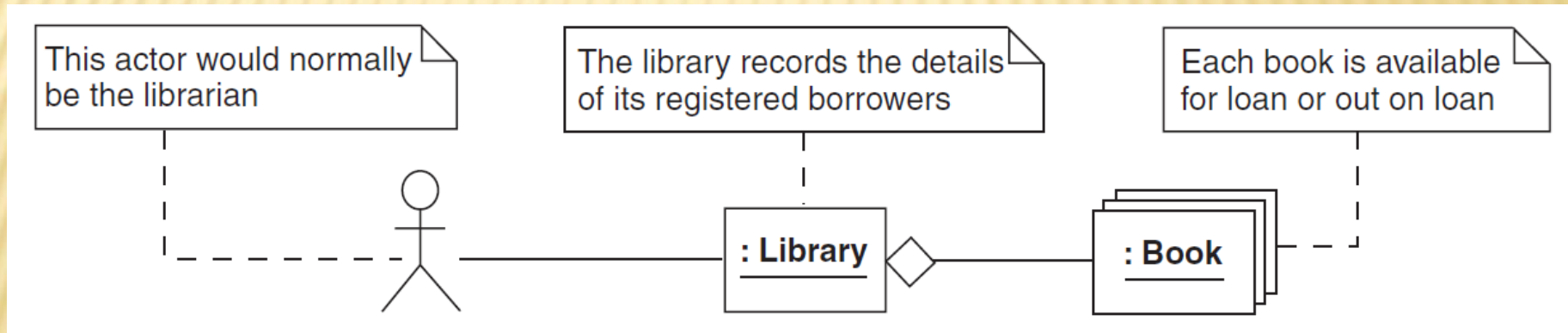
A book available for loan is recorded as out on loan to a borrower. The book is not available for loan.

6. Return one book

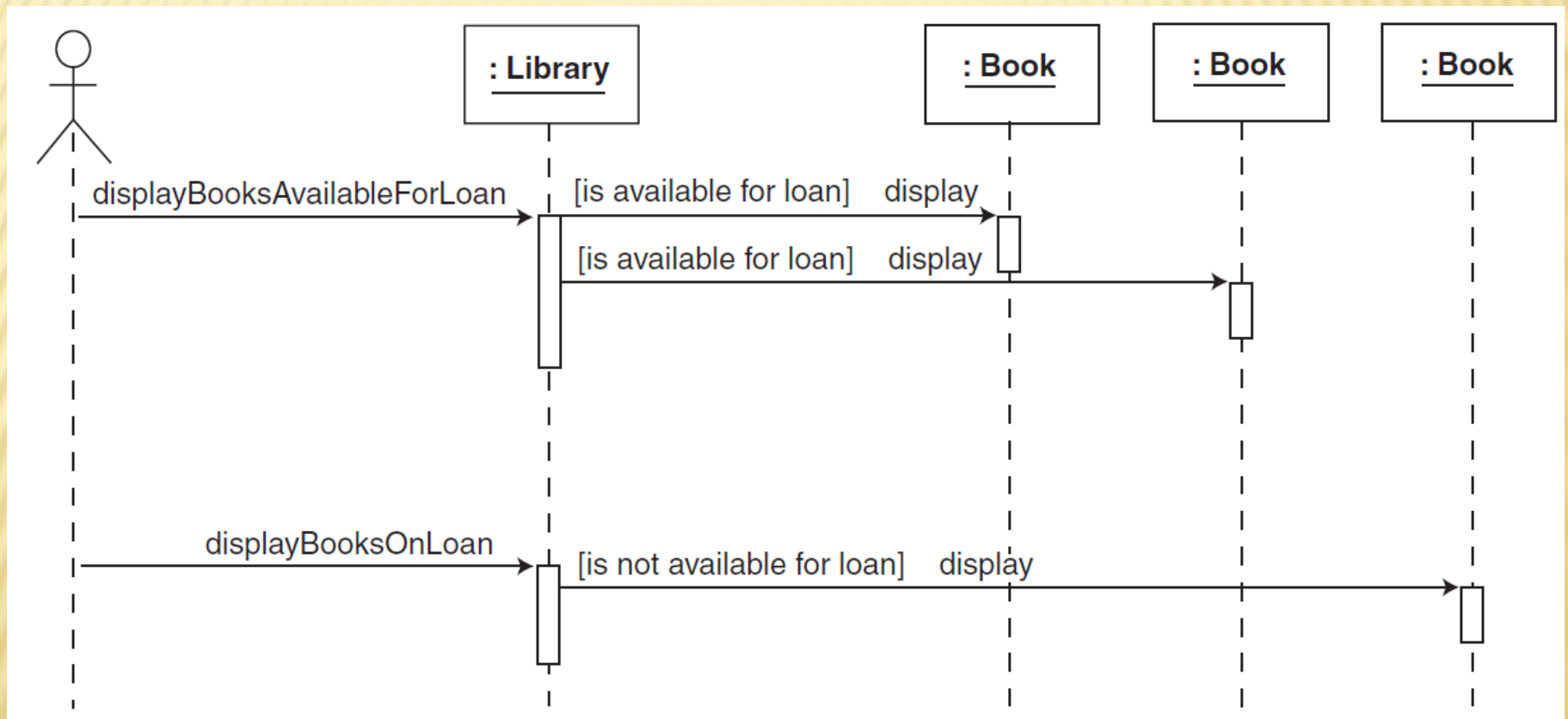
A book on loan to a borrower is recorded as not out on loan to that borrower. The book is available for loan.

ANALYSIS - OBJECT DIAGRAM

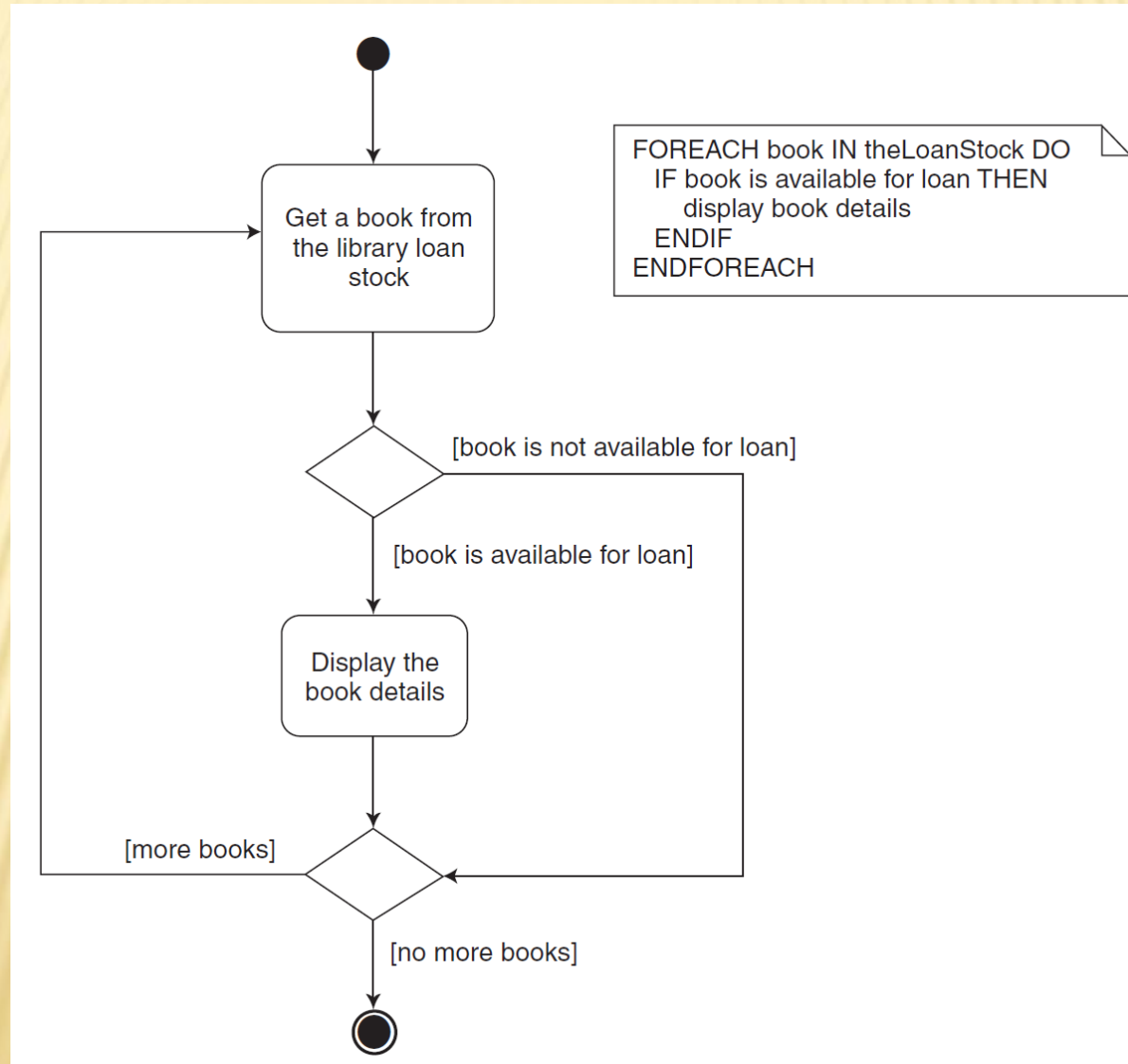
- The librarian and borrower objects are not part of the system.
- Each **Book** is part of the Library



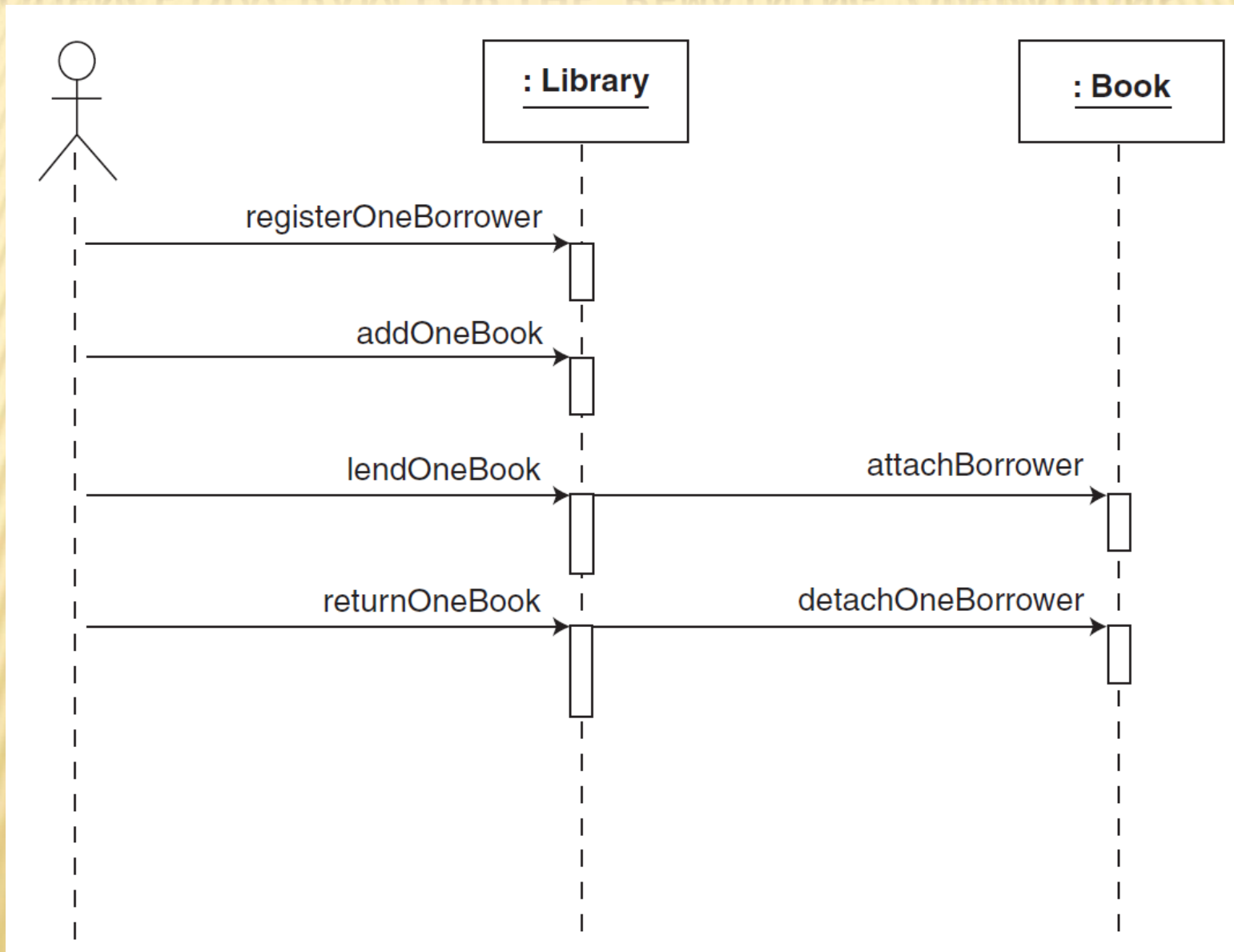
A SEQUENCE DIAGRAM FOR THE DISPLAYBOOKSAVAILABLEFORLOAN OPERATION



A ACTIVITY DIAGRAM FOR THE DISPLAYBOOKSAVAILABLEFORLOAN OPERATION



A SEQUENCE DIAGRAM FOR THE REMAINING OPERATIONS



A COLLABORATION DIAGRAM FOR THE LENDONEBOOK OPERATION

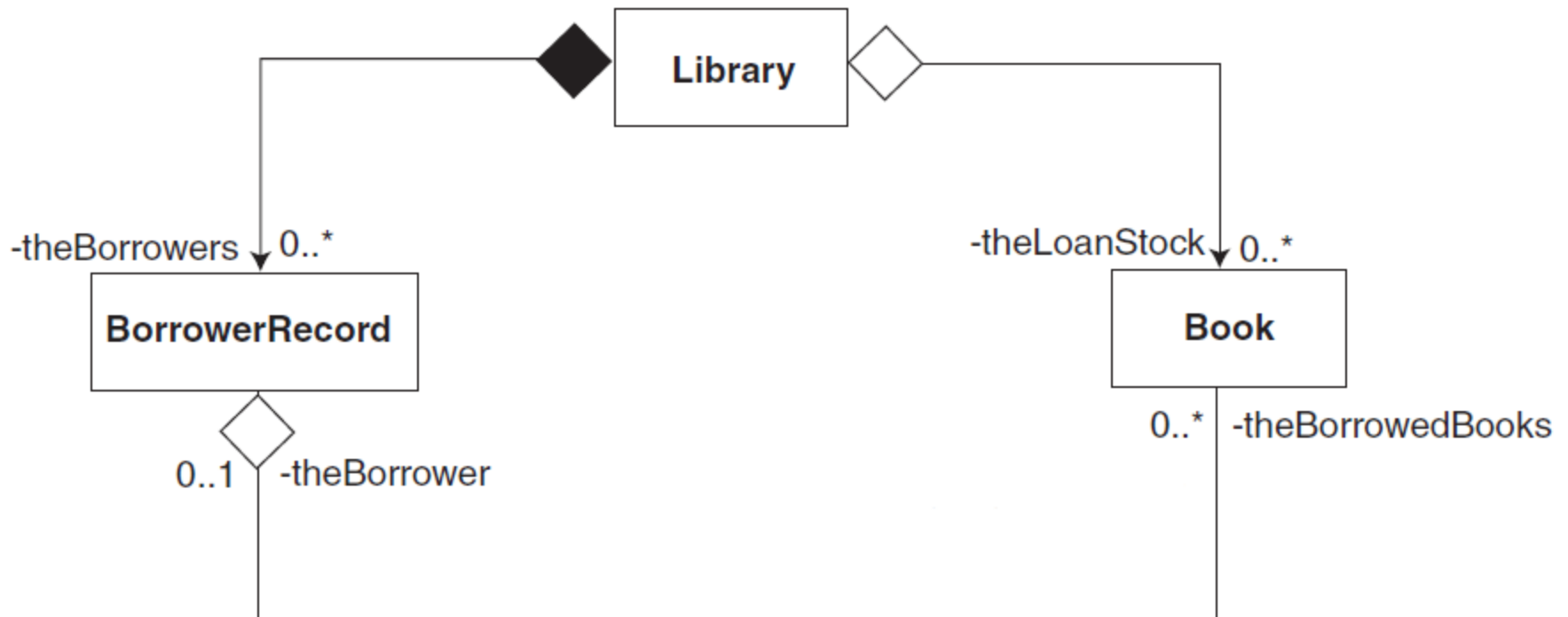
If a book is not attached to a borrower then it is available for loan.
Otherwise it is out on loan.

This book is now attached to a borrower while it is out on loan.
It will be detached from that borrower on returning it.



DESIGN - CLASS DIAGRAM

- ✗ The Library needs to have a record of each registered borrower



THE IMPACT OF INTRODUCING THE BORROWERRECORD CLASS

- ✗ Register one borrower:
 - + Establish a suitably initialized **BorrowerRecord** object
- ✗ Add one book:
 - + Set **theBorrower** in the **Book** to **NULL**
- ✗ Display books available:
 - + Each **Book** displayed has **theBorrower** equal for loan to **NULL**
- ✗ Display books out:
 - + Each **Book** displayed has **theBorrower** not on loan equal to **NULL**
- ✗ Lend one book:
 - + Attach a **BorrowerRecord** to a **Book** and vice versa
- ✗ Return one book:
 - + Detach a **BorrowerRecord** from a **Book** and vice versa

CLASS DIAGRAM DETAILS

Library

-theName: String

+Library(String aName)

+registerOneBorrower(String aBorrowerName)

+addOneBook(Book aBook)

+displayBooksAvailableForLoan()

+displayBooksOnLoan()

+lendOneBook(String aCatalogueNumber, String aBorrowerName)

+returnOneBook(String aCatalogueNumber)

CLASS DIAGRAM DETAILS

Book

-theCatalogueNumber: String
-theAuthor: String
-theTitle: String

+Book(String aCatalogueNumber, String aTitle, String anAuthor)
+attachBorrower(BorrowerRecord aBorrower)
+detachBorrower()
+display()

BorrowerRecord

-theName: String

+BorrowerRecord(String aName)
+attachBook(Book aBook)
+detachBook(Book aBook)