

Software Engineering - Blatt 5

Abgabegruppe 14

Leonid Antipov	583118
Alyona Buyukli	526898
Karol Czopek	581035
Levin Palm	581323

Contents

0.1	Class Index	1
0.1.1	Class List	1
0.2	Class Documentation	1
0.2.1	book Class Reference	1
0.2.1.1	Detailed Description	1
0.2.1.2	Constructor & Destructor Documentation	1
0.2.1.3	Member Function Documentation	2
0.2.2	library Class Reference	3
0.2.2.1	Detailed Description	3
0.2.2.2	Constructor & Destructor Documentation	4
0.2.2.3	Member Function Documentation	4
0.2.3	user Class Reference	7
0.2.3.1	Detailed Description	8
0.2.3.2	Constructor & Destructor Documentation	8
0.2.3.3	Member Function Documentation	8
	Index	11

0.1 Class Index

0.1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

book	1
library	3
user	7

0.2 Class Documentation

0.2.1 book Class Reference

```
#include <book.h>
```

Public Member Functions

- [book](#) (std::string title, std::string author, unsigned long isbn)
- bool [borrow](#) (unsigned long userID)
- bool [bringBack](#) ()
- std::string [getTitle](#) () const
- std::string [getAuthor](#) () const
- unsigned long [getISBN](#) () const
- bool [isRetired](#) () const
- bool [isBorrowed](#) () const
- unsigned int [getTimesBorrowed](#) () const
- unsigned long [getBorrowingUserID](#) () const
- std::string [toString](#) ()
- bool [operator<](#) (const [book](#) &other) const
- bool [operator==](#) (const [book](#) &other) const

0.2.1.1 Detailed Description

[book](#) provides the representation of a book each book has a title, an author and an isbn number the isbn number is supposed to be unique

0.2.1.2 Constructor & Destructor Documentation

0.2.1.2.1 [book](#)()

```
book::book (
    std::string title,
    std::string author,
    unsigned long isbn )
```

Create a new book object

Parameters

<i>title</i>	The title of the book
<i>author</i>	The author of the book
<i>isbn</i>	The unique isb number of the book

0.2.1.3 Member Function Documentation**0.2.1.3.1 borrow()**

```
bool book::borrow (
    unsigned long userID )
```

borrow this book this will set the internal variables to indicate that this book is borrowed this will also update the times this is borrowed and retieres the book if needed

Parameters

<i>userID</i>	the ID of the user who wants to borrow this book
---------------	--

Returns

true if the borrowing was successful, false if the book is already borrowed or retired

0.2.1.3.2 bringBack()

```
bool book::bringBack ( )
```

bring this book back this will set the internal status to not borrowed will also manage the retirement process

Returns

true on success, false if the book is currently not borrowed

0.2.1.3.3 operator<()

```
bool book::operator< (
    const book & other ) const [inline]
```

the < operator is valid on a book a book is smaller than a book if its isb number is smaller

0.2.1.3.4 operator==()

```
bool book::operator== (
    const book & other ) const [inline]
```

the == operator is valid on a book a book is equal to another book if their isb numbers are equal

0.2.1.3.5 toString()

```
std::string book::toString ( )
```

Creates a string representation of this book

Returns

String in form [author]: [title] ([isbn])

The documentation for this class was generated from the following files:

- book.h
- book.cpp

0.2.2 library Class Reference

```
#include <library.h>
```

Public Member Functions

- `library ()`
- unsigned int `getBookCount () const`
- unsigned int `getUserCount () const`
- unsigned int `registerUser (std::string name, std::string address)`
- `user * findUserById (unsigned int userID)`
- void `wishForABook (user wishingUser, book bookWishedFor)`
- void `addBookToLibrary (book bookToAdd)`
- bool `borrowBook (book *bookToBorrow, user *borrowingUser)`
- bool `returnBook (book *bookToReturn, user *borrowingUser)`
- std::vector< book > `getAllBooks ()`
- std::vector< user > `getAllUsers ()`
- `user * getUserWhoBorrowsBook (book bookToCheck)`
- unsigned int `getTimesBorrowed (book bookToCheck)`
- unsigned int `getTimesWishedFor (book bookToCheck)`

0.2.2.1 Detailed Description

library provides the main interface. This contains all the books and users. It also allows the API-user to create and manage users as well as books.

0.2.2.2 Constructor & Destructor Documentation

0.2.2.2.1 library()

```
library::library ( )
```

creates a new library with 0 registered users and 0 books

0.2.2.3 Member Function Documentation

0.2.2.3.1 addBookToLibrary()

```
void library::addBookToLibrary (
    book bookToAdd )
```

This method adds a given book to the library. Books that are already in the library will be ignored

Parameters

<i>bookToAdd</i>	the book that should be added to the library
------------------	--

0.2.2.3.2 borrowBook()

```
bool library::borrowBook (
    book * bookToBorrow,
    user * borrowingUser )
```

This method allows a user to borrow book a from the library The method checks if the user is registerd and if the book is available in this library.

Parameters

<i>bookToBorrow</i>	A pointer to the book that should be borrowed
<i>borrowingUser</i>	A pointer to the user that wants to borrow the book

Returns

true if successful, false if book not available or user not registered

0.2.2.3.3 findUserById()

```
user * library::findUserById (
    unsigned int userID )
```

Search by the user id for a user in the system

Parameters

<i>userID</i>	The user id that should be searched
---------------	-------------------------------------

Returns

A pointer to the user with the given id, or nullptr if no user with this id exists

0.2.2.3.4 getAllBooks()

```
std::vector< book > library::getAllBooks ( )
```

Returns

a hard copy of the list of books that are currently available in the library

0.2.2.3.5 getAllUsers()

```
std::vector< user > library::getAllUsers ( )
```

Returns

a hard copy of the list of users that are currently registered in the library

0.2.2.3.6 getTimesBorrowed()

```
unsigned int library::getTimesBorrowed (
    book bookToCheck )
```

Parameters

<i>bookToCheck</i>	the book to check this on
--------------------	---------------------------

Returns

the number of times the book has been borrowed

0.2.2.3.7 `getTimesWishedFor()`

```
unsigned int library::getTimesWishedFor (
    book bookToCheck )
```

Parameters

<i>bookToCheck</i>	The book to check this on
--------------------	---------------------------

Returns

the number of times individual users wished for this book

0.2.2.3.8 `getUserWhoBorrowsBook()`

```
user * library::getUserWhoBorrowsBook (
    book borrowedBook )
```

Given a book this will return which user currently borrows this book

Parameters

<i>borrowedBook</i>	the book to check this on
---------------------	---------------------------

Returns

a pointer to the user that currently borrows this book

0.2.2.3.9 `registerUser()`

```
unsigned int library::registerUser (
    std::string name,
    std::string address )
```

This method registers a new user in this library

Parameters

<i>name</i>	The name the new user has
<i>address</i>	The address the new user has

Returns

the ID the new user will have

0.2.2.3.10 returnBook()

```
bool library::returnBook (
    book * bookToReturn,
    user * borrowingUser )
```

This method allows a user to give a borrowed book back to the library. The method checks if the user is registered and if the book belongs to this library.

Parameters

<i>bookToReturn</i>	A pointer to the book that the user wants to hand in
<i>borrowingUser</i>	A pointer to user that currently borrows this book

Returns

true if successful, false if book does not belong to this library, or book is not being borrowed or user not registered

0.2.2.3.11 wishForABook()

```
void library::wishForABook (
    user wishingUser,
    book bookWishedFor )
```

This method allows a user to wish for a certain book to be added to the library.

Parameters

<i>wishingUser</i>	The user that wishes for the book
<i>bookWishedFor</i>	The book that the user wishes for

The documentation for this class was generated from the following files:

- library.h
- library.cpp

0.2.3 user Class Reference

```
#include <user.h>
```

Public Member Functions

- **user** (unsigned int userID, std::string name, std::string address)
- unsigned int **getUserID** () const
- std::string **getName** () const
- std::string **getAddress** () const
- bool **borrowBook** (book *bookToBorrow)
- bool **returnBook** (book *bookToReturn)
- std::vector< book > **getBorrowedBooks** ()
- bool **operator==** (const user &other) const

0.2.3.1 Detailed Description

this class provides all the information that is needed to manage a user each user has a name, address information and a systemwide unique id the user class also manages the books the user currently borrows

0.2.3.2 Constructor & Destructor Documentation

0.2.3.2.1 user()

```
user::user (
    unsigned int userID,
    std::string name,
    std::string address )
```

create a new user with a userID

Parameters

<i>userID</i>	the unique id (for uniqueness this should be assigned by registerUser())
<i>name</i>	the real name of this user
<i>address</i>	the address information of this user

0.2.3.3 Member Function Documentation

0.2.3.3.1 borrowBook()

```
bool user::borrowBook (
    book * bookToBorrow )
```

This method allows this user to borrow a book

Parameters

<i>bookToBorrow</i>	A pointer to the book this user wants to borrow
---------------------	---

Returns

true on success, false if this user cant borrow this book

0.2.3.3.2 getBorrowedBooks()

```
std::vector< book > user::getBorrowedBooks ( )
```

Returns

a list containing all the books that this user borrows

0.2.3.3.3 operator==()

```
bool user::operator== (
    const user & other ) const [inline]
```

the == operator is valid on users a user is valid to another user if their ids are identical

0.2.3.3.4 returnBook()

```
bool user::returnBook (
    book * bookToReturn )
```

this method allows the user to return a book to the library it is verified that he borrowed that book and that this book belongs to this library

Parameters

<i>bookToReturn</i>	the book that should be returned
---------------------	----------------------------------

Returns

true on success, false if this book can not be returned because it does not belong here or is not borrowed

The documentation for this class was generated from the following files:

- user.h
- user.cpp

Index

addBookToLibrary
library, 4

book, 1
book, 1
borrow, 2
bringBack, 2
operator<, 2
operator==, 2
toString, 3

borrow
book, 2

borrowBook
library, 4
user, 8

bringBack
book, 2

findUserById
library, 4

getAllBooks
library, 5

getAllUsers
library, 5

getBorrowedBooks
user, 8

getTimesBorrowed
library, 5

getTimesWishedFor
library, 5

getUserWhoBorrowsBook
library, 6

library, 3
addBookToLibrary, 4
borrowBook, 4
findUserById, 4
getAllBooks, 5
getAllUsers, 5
getTimesBorrowed, 5
getTimesWishedFor, 5
getUserWhoBorrowsBook, 6
library, 4
registerUser, 6
returnBook, 6
wishForABook, 7

operator<
book, 2

operator==

book, 2
user, 9

registerUser
library, 6

returnBook
library, 6
user, 9

toString
book, 3

user, 7
borrowBook, 8
getBorrowedBooks, 8
operator==, 9
returnBook, 9
user, 8

wishForABook
library, 7