



Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo de Projeto André Sá

Library Manager

- library; Library

+ Save(): void
 + SaveAs(filename: String): void
 + load(filename: String): void
 + importFile(filename: String): void
 + doAdvanceDate(increment: int): void
 + doDisplayDate(): int
 + doRequestWork(userID: int, workID: int): void
 + doReturnWork(userID: int, workID: int): void
 + doPayFine(userID: int): void
 + doRegisterUser(name: String, email: String): void
 + doShowUser(userID: int): String
 + doShowUserNotifications(userID: int): List<String>
 + doShowUsers(): List<String>
 + doChangeWorkInventory(workID: int, change: int): void
 + doDisplayWork(workID: int): String
 + doDisplayWorks(): List<String>
 + doDisplayWorksByCreator(creator: String): List<String>
 + doPerformSearch(prompt: String): List<String>

ImportFileException

- <<final>> SerialVersionUID: long
 - <<final>> ERROR_MESSAGE: String

MissingFileAssociationException

- <<final>> SerialVersionUID: long

UnavailableFileException

- <<final>> SerialVersionUID: long
 - <<final>> filename: String
 + getFilename(): String

UnrecognizedEntryException

- <<final>> SerialVersionUID: long
 - <<final>> entrySpecification: String
 + getEntrySpecification(): String

ImportFileException

MissingFileAssociationException

UnavailableFileException

UnrecognizedEntryException

BorrowingRuleFailedException

FileOpenFailedException

NoSuchCreatorException

NoSuchUserException

NoSuchWorkException

UserIsActiveException

UserRegistrationFailedException

WorkNotBorrowedByUserException

Exception

Creator
- <<final>> _name: String + Creator(name: String) + addWork(work: Work): void + rmWork(workID: int): void + hasWorks(): boolean

Book
- <<final>> Authors: List<Creator> - <<final>> ISBN: String DVD - <<final>> Director: Creator - <<final>> ISAC Reg: String

Request
- _user: User - _work: Work - _requestDate: int - _dueDate: int

<<abstract>> Behavior
- nConcurrentRequests: int - dueDateSingleCopy: int - dueDateFewCopies: int - dueDateManyCopies: int

NonCompliant
Normal
Compliant

<<abstract>> Work
- <<final>> _title: String - <<final>> _genre: String - <<final>> _id: int - _inventory: int - _available: int - _price: int + changeInventory(num: int): void + requestWork(): void + returnWork(): void + addNotificationBorrow(user: User): void + addNotificationAvailable(user: User): void + rmNotificationBorrow(userID: int): void + rmNotificationAvailable(userID: int): void + work(title: String, genre: String, id: int, inv: int, price: int)

User
- _name: String - _email: String - <<final>> _id: int - _status: boolean - _behavior: Behavior - _requestResults: List<boolean> - _fine: int - _notifications: List<String> *1 + addWork(work: Work): void + rmWork(workID: int): void + addRequestResult(onTime: boolean): void + addRequest(request: Request): void + payFine(): void + updateStatus(): void + updateBehavior(): void + addNotification(notification: String): void + showAndEmptyNotifications(): List<String>

*1 rm Creator(name: String): void
2

*1 User(name: String, email: String, id: int)

Library
- <<final>> serialVersionUID: long - _date: int + addDVD(title: String, genre: String, price: int, inventory: int, creator: String, ISAC: String): void + addBook(title: String, genre: String, price: int, inventory: int, creator: List<String>, ISBN: String): void + addUser(name: String, email: String): void + addRequest(userID: int, workID: int): void + checkRequest(userID: int, workID: int): boolean + addNotificationBorrow(userID: int, workID: int): void + addNotificationAvailable(userID: int, workID: int): void + rmWork(workID: int): void + rmUser(userID: int): void + rmRequest(userID: int, workID: int): void + rmNotificationBorrow(userID: int, workID: int): void + rmNotificationAvailable(userID: int, workID: int): void + showWork(id: int): String + showAllWorks(): List<String> + showUser(id: int): String + showUserNotifications(id: int): List<String> + showAllUsers(): List<String> + showCreatorWorks(creator: String): List<String> + showDate(): int + search(prompt: String): List<String> + payFine(userID: int): void + changeInventory(workID: int, invChange: int): void + advanceDate(increment: int): void + importFile(filename: String): void *2