UML class diagram. Transcription of labels and text:

bci (top package)
- Library Manager
- Library (─library)
- ─ rules → Rule Set

bci.util (package, top right)
- Rule Set

bci.util.work (package, left)
- Creator
- Work
  - ─ autors *
  - ─ works *
  - ─ category
- Category
- Book
- DVD
- Fiction Category
- SciTech Category
- Reference Category

bci.util (center)
- Request (─work, ─user)
- Notification
  - ─ notification *
  - Request Notification
  - Return Notification

bci.util.user (package)
- User (─status)
- Status
  - Active Status
  - Suspended Status
- Classification (─classification)
  - Normal Classification
  - Irresponsible Classification
  - Combiant Classification

bci.exceptions (package)
- Import File Exception
- Missing File Exception
- Unavailable File Exception
- Unrecognized Entry Exception
- Work Exception
- User Exception
- Exception

Todas as classes, exceto Library Manager, implementam Serializable. Os seus serial numbers foram omitidos.

Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o meu grupo de projeto (apenas eu). André Cadite

LibraryManager

- _library : Library
- __associatedFilename : String

+ save() : void
+ saveAs(filename : String) : void
+ load(filename : String) : void
+ importFile(filename : String) : void
+ exportFile() : void
+ getCurrentDate() : int
+ advanceDate(days : int) : void
+ registerUser(name : String, email : String) : void
+ showUser(uid : int) : String
+ showUserNotifications(uid : int) : String
+ showUsers() : String
+ payAllFines(uid : int) : void
+ ~~getTotalDebt(uid : int)~~
+ showWork(wid : int) : String
+ showWorks() : String
+ showWorksByCreator(name : String) : String
+ changeWorkInventory(ammount : int) : void
+ searchWorks(term : String) : String
+ requestWork(uid : int, wid : int) : int
+ addReturnNotification(uid : int, wid : int) : void
+ returnWork(uid : int, wid : int) : int
+ payFine(uid : int, wid : int) : void

Library

- _users : ArrayList<User>	- _rules : RuleSet
- _works : ArrayList<Work>	
- _creators : HashSet<Creator>	
- _requests : ArrayList<Request>	
- _currentDate : int	

~ importFile(filename : String) : void
~ exportFile() : void
+ getCurrentDate() : int
+ advanceDate(days : int) : void
+ registerUser(name : String, email : String) : void
+ showUser(uid : int) : String
+ showUserNotifications(uid : int) : String
+ showUsers() : String
~ payAllFines(uid : int) : void
+ showWork(wid : int) : String
+ showWorks() : String
+ showWorksByCreator(name : String) : String
+ changeWorkInventory(ammount : int) : void
+ searchWorks(term : String) : String
+ requestWork(uid : int, wid : int) : int
+ addReturnNotification(uid : int, wid : int) : void
+ returnWork(uid : int, wid : int) : int
+ payFine(uid : int, wid : int) : void

User

- _uid: int
- _name: String
- _email: String
- _status: Status Status
- _classification: Classification
- _fines: HashMap<Integer, Integer>
- _notifications: ArrayList<Notification>

+ addFine(wid: int, value: int): void
+ payFine(wid: int): void
+ payAllFines(): void
+ getFine(wid: int): int
+ getTotalDebt(): int
+ showNotifications(): String

Work

- _wid: int
- _copies: int
- _stock: int
- _title: String
- _price: int
- _category: Category
- _creators: Creator[]
- _request_interested_users: HashSet<User>
- _return_interested_users: HashSet<User>

+ incrementStock(): void
+ decrementStock(): void
+ addCopies(value: int): void
+ addRequestInterestedUser(uid: int): void
+ removeRequestInterestedUser(wid: int): void
+ addReturnInterestedUser(user: User): void

Creator

- _name: String
- _works: TreeSet<Work>

+ appendWork(newWork: Work): void
+ removeWork(oldWork: Work): void
+ containsWorks(): boolean

Request

- _user: User
- _work: Work
- _deadline: int

+ isOverdue(date: int): boolean

Rule Set

- _rules: int[]
- _maxRequests: int[]
- _deadlines: int[][]
- _maxWorkPrice: int[]
- _fineAmmount: int

+ canBorrow(user: User, work: Work): boolean
+ getDeadline(copies: int, status: Status): int

DVD

- _IGAC: String

Book

- _ISBN: String

<< interface >>
Category

−codename : String

bci.util.work

<< interface >>
Notification

−work : Work

bci.util.user

<< interface >>
Status

−s : int

<< interface >>
Classification

−c : int