به نام خدا

تمرین سری سوم

مائده بادان فیروز - ۹۸۲۲۲۰۰۹

بهار ۱۴۰۲

مقدمه:

هدف طراحی یک کتابخانه‌ی تقریبا جامع، با قابلیت نگهداری مخزن کتاب، لیست کاربران و لیست کتابدارها است. در این برنامه به کاربر قابلیت تغییر نام کاربری و گذرواژه، جست‌وجو در کتاب‌ها و امانت گرفتن یا پس دادن کتاب را می‌دهد و به کتابدارها قابلیت اضافه کردن، حذف کردن و یا به‌روزرسانی کردن هر کدام از کاربرها یا کتابدارها یا کتاب‌ها را می‌دهد.

طراحی و پیاده سازی:

کلاس BookEntity صرفا برای اینکه بتوانیم برای به روزرسانی راحتتر عمل کنیم ایجاد شده است. کار یک object از این نوع این است که صرفا هر تعداد از اطلاعاتی که قرار است تغییر داده شود را در خود نگه دارد. مشابها AccountEntity را داریم.

BookEntity
- name: String = "" - author: String = "" - yearOfPublish: int = 0 - isbn: String = ""
+ BookEntity()

Book
- name: String - author: String - yearOfPublish: int - isbn: String - isInRent: Boolean = false
+ BookEntity(name: String, author: String, yearOfPublish: int, isbn: String) + getName(): String +setName(name: String): void + getAuthor(): String +setAuthor(author: String): void + getYearOfPublish(): String +setYearOfPublish(name: String): Integer + getIsbn(): String + setIsbn(isbn: String): void + getIsInRent(): boolean + setIsInRent(): void + rentBook(): Book + returnBook(): void + update(bookEntity: BookEntity): Book + toString(): String

AccountEntity
- username: String = "" - password: String = ""
+ AccountEntity()

کلاس AccountEntity صرفا برای اینکه بتوانیم برای به روزرسانی راحتتر عمل کنیم ایجاد شده است.

Account
- username: String - password: String
+ Account(username: String, password: String) + getUsername(): String + setUsername(username: String): void + getUsername(): String + setPassword(password: String) + update(accountEntity: AccountEntity): Account + toString(): String

User
- booksInRent: ArrayList<Book>
+ User(username: String, password: String) + getBooksInRentByUser(): Book[] + rentBook(book: Book): void + returnBook(book: Book): void + update(accountEntity: AccountEntity): Account + doesRentBook(book: Book): boolean

Librarian
+ Librarian(username: String, password: String) + update(accountEntity: AccountEntity): Account

نکته: به قابلیت‌های بیشتر برای کتابدار در منوی تابع main رسیدگی شده. عملیات‌ها با توابعی که در Library نوشته شده انجام می‌شوند و صرفا وقتی کاربر به عنوان کتابدار وارد شده به منویی که اجازه‌ی عملیات‌های خاص را دارد هدایت می‌شود.

Library
- listOfBooks: ArrayList<Book> - listOfUsers: ArrayList<User> - listOLibrarians: ArrayList<Librarian> - bookCounter: HashMap<String, Integer>
+ addBook(book: Book): void + removeBook(book: Book): boolean + updateBook(book: Book, bookEntity: BookEntity): Book + searchBook(bookEntity: BookEntity): Book[] + doesBookExist(isbn: String):Boolean + borrowBook(user: User, book:Book): boolean + returnBook(user:User, book: Book): boolean + increaseBook(isbn: String): void + decreaseBook(isbn: String): void + getBookStatistic(bookName: String): String[] + addUser(user: User): void + removeUser(user: User): boolean + updateUser(user: User, accountEntity: AccountEntity): User + searchUser(username: String): User + doesUserExist(username: String): Boolean + addLibrarian(librarian: Librarian): void + removeLibrarian(librarian: Librarian): boolean + updateLibrarian(librarian: Librarian, accountEntity: AccountEntity) + searchLibrarian(username: String): Librarian + doesLibrarianExist(username: String): boolean

سنجش و ارزیابی:

در کلاس تست سعی شد از هر کلاسی یک نمونه ساخته شود و برای هر کدام برخی توابع که کارشان پیچیدگی بیشتری دارد امتحان شوند. علاوه بر آن در زمان نوشتن main هم توابع بقیه کلاس‌ها استفاده و بعضا debug شدند. بیشتر در حین نوشتن منوهای main اشکال‌ها پیدا و رفع شدند.

نتیجه‌گیری:

پروژه به این سنگینی برای یک هفته زیاد بود! :)))))