

**دانشکده ریاضی و علوم کامپیوتر**

**برنامه‌سازی پیشرفته و کارگاه**

**تمرین Exception Handling**

استاد درس

دکتر مهدی قطعی

استاد دوم

بهنام یوسفی مهر

نگارش

یونس کاظمی مقدم

اسفند ۱۴۰۳

**فهرست**

[مقدمه 3](#_Toc192363137)

[شرح کلی تمرین و معرفی کلاس‌ها 4](#_Toc192363138)

[کلاس Book 4](#_Toc192363139)

[فیلد‌ها 4](#_Toc192363140)

[متد‌ها 4](#_Toc192363141)

[کلاس Library 5](#_Toc192363142)

[فیلد‌ها 5](#_Toc192363143)

[متد‌ها 5](#_Toc192363144)

[کلاس های Exception 6](#_Toc192363145)

[کلاس Main 7](#_Toc192363146)

[چند نکته 8](#_Toc192363147)

مقدمه

این تمرین قراره کمک کنه که **Exception Handling** رو توی جاوا بهتر یاد بگیرین. می‌خوایم یه **سیستم مدیریت کتابخونه** بسازیم که توی اون بتونین کتاب اضافه کنین، کتاب قرض بگیرین و کتاب رو پس بدین. ولی خب، همیشه همه چی خوب پیش نمی‌ره؛ ممکنه بخواین یه کتابی رو بگیرین که تو کتابخونه نیست، یا کتابخونه خالی باشه و بخواین لیست کتاب‌ها رو داشته باشین! اینجاست که باید با **Custom Exceptions** و **try-catch** تو جاوا کار کنین تا بتونین برنامه‌ای بنویسین که کرش نکنه و پیام‌های درستی به کاربر بده.

راستی حتما قبل از شروع به نوشتن این تمرین، قسمت **«چند نکته»** در پایان این داکیومنت رو هم بخونین.

شرح کلی تمرین و معرفی کلاس‌ها

در این قسمت، کلاس‌های مختلف پروژه و نقش هر کدوم رو توضیح می‌دیم.

کلاس Book

این کلاس مسئول نگهداری اطلاعات مربوط به یه کتابه.

فیلد‌ها

**private String title:** عنوان کتاب

**private int pageCount:** تعداد صفحات کتاب

متد‌ها

**public Book(String title, int pageCount):** کانستراکتور کلاس هست که یک کتاب رو با نام title و تعداد صفحات pageCount ایجاد میکنه.

دقت کنین که اگه نام کتاب خالی بود[[1]](#footnote-1) باید یک IllegalArgumentException با پیام *Title cannot be empty* رو throw کنین.

همچنین اگه تعداد صفحات کتاب صفر یا منفی بود باید یک IllegalArgumentException با پیام *Page count must be positive* رو throw کنین.

در صورتی که نام و تعداد صفحات کتاب مشکلی نداشتن، فیلد‌های مربوطه، با این مقادیر مقداردهی می‌شن.

**تمام فیلد‌های private باید getter با نام مناسب داشته باشن[[2]](#footnote-2).**

**public String toString():** متد toString باید Override بشه. به این شکل که اگه کتابی با عنوان Sample Book دارای 100 صفحه هست و متد toString روی اون صدا زده بشه، باید خروجی متد مطابق زیر باشه:

Sample Book (100 pages)

کلاس Library

این کلاس به عنوان مدیریت‌کننده‌ی کتاب‌ها عمل می‌کنه و تمام کتاب‌های موجود تو کتابخونه رو نگه می‌داره.

فیلد‌ها

**private ArrayList<Book> books:** کتاب‌های موجود در کتابخانه

متد‌ها

**public Library():** کانستراکتور کلاس هست که توی اون آبجکت books ساخته میشه (new میشه).

**public void addBook(Book book) throws InvalidBookException:** این متد سعی میکنه کتاب book رو به لیست books اضافه کنه. اگه آبجکت book برابر null باشه باید یک InvalidBookException با پیام *Book should not be null.* رو throw کنین.

در نهایت، کتاب book رو به لیست books اضافه کنین.

**private Book findBook(String title):** این متد به دنبال کتابی با عنوان title در لیست books می‌گرده. اگه چنین کتابی وجود داشت، اون کتاب رو برمی‌گردونه و در غیراینصورت null رو بر می‌گردونه.

**public void borrowBook(String title) throws BookNotFoundException, EmptyLibraryException:** این متد سعی میکنه کتابی با نام title رو امانت بده.

در ابتدا اگه لیست books خالی بود باید یک EmptyLibraryException با پیام *Library is empty, no books to borrow* رو throw کنین.

سپس با صدا زدن متد findBook به دنبال کتابی با عنوان title در لیست books می‌گردیم. اگه چنین کتابی وجود نداشت باید یک BookNotFoundException با پیام *Book with title " + title + " not found.* رو throw کنین.

در نهایت اگه مشکلی نبود، پیامی مبنی بر موفقیت‌آمیز بودن امانت گرفتن کتاب به همراه مشخصات کتاب چاپ می‌شه.

**public void returnBook(String title) throws BookNotFoundException:** این متد سعی می‌کنه کتابی که دارای عنوان title هست رو برگردونه.

در ابتدا با صدا زدن متد findBook بررسی میشه آیا کتابی با این عنوان در لیست books وجود داره یا نه. اگه وجود نداشت باید یک BookNotFoundException با پیام *Cannot return. Book with title " + title + " not found.* رو throw کنین.

اگر چنین کتابی وجود داشت، پیامی مبنی بر موفقیت‌آمیز بودن برگرداندن کتاب به همراه مشخصات کتاب چاپ می‌شه.

**public void listBooks() throws EmptyLibraryException:** این متد سعی می‌کنه اطلاعات تمام کتاب‌های موجود در کتابخونه رو چاپ کنه.

در ابتدا اگه لیست books خالی بود باید یک EmptyLibraryException با پیام *Library is empty.* رو throw کنین.

اگه books خالی نبود، ابتدا پیام *Books in the library:* چاپ می‌شه و در خطوط بعدی کتاب‌های موجود در کتابخونه چاپ میشن.

**تمام فیلد‌های private باید getter با نام مناسب داشته باشن.**

کلاس های Exception

در این تمرین سه کلاس Exception اختصاصی داریم: BookNotFoundException و EmptyLibraryException و InvalidBookException.

توجه کنین که هر یک از این کلاس‌ها باید در یک فایل جداگانه نوشته بشن.

همچنین هر یک از این کلاس‌ها از کلاس Exception ارث‌بری میکنن.

در تمامی این کلاس‌ها باید کانستراکتوری که ورودی آن تنها یک String هست نوشته شه (ترجیحا نام این String رو message بذارین). در بدنه این کانستراکتور، کانستراکتور والد با آرگومان message، فراخوانی می‌شه.

کلاس Main

نمونه

چند نکته

همونطور که در طول تمرین هم گفته شد، در همه کلاس‌ها باید تمام فیلد‌های private دارای getter با نام مناسب باشن.

1. یک title رو میگیم خالی هروقت برابر null باشه یا یک رشته تهی باشه. [↑](#footnote-ref-1)
2. منظور از نام مناسب این هست که اگه مثلا اسم یه فیلد sample هست، باید اسم متد getter اون getSample باشه. [↑](#footnote-ref-2)