



## شرح مسئله

در این مسئله قرار است قواعد امانت گیری مستندات یک کتابخانه پیاده سازی شود. در این کتابخانه دانشجویان و اساتید می توانند انواع مستندات را از کتابخانه امانت بگیرند. قرار است به کمک این سیستم بتوانیم امانت دادن مستندهای موجود در کتابخانه را سازماندهی کنیم و در هر زمان بتوانیم فهرست کتابهای امانت داده شده، افرادی که این کتابها را امانت گرفته اند و مدت زمان دیرکرد هر کتاب را بازبایی نماییم. اعضای کتابخانه می توانند علاوه بر کتابهای عادی، کتابهای مرجع و مجلات را نیز از کتابخانه امانت بگیرند. در هر لحظه، یک استاد می تواند حداکثر ۵ مستند و یک دانشجو حداکثر ۲ مستند را در امانت داشته باشند. از هر یک از مستندات ممکن است چند نسخه در کتابخانه موجود باشد. در این صورت بیش از یک نفر می توانند آن مستند را همزمان به امانت بگیرند. بدیهی است که اگر تمام نسخه ها در امانت باشند آن مستند برای امانت گیری در دسترس نخواهد بود.

مهلت تحویل کتابها و جریمه دیرکرد انواع مستندات طبق جدول زیر تعیین می شود.

| مهلت تحویل (روز) | کتاب عادی | کتاب مرجع | مجله |
|------------------|-----------|-----------|------|
|                  | ۱۰        | ۵         | ۲    |

اعضای کتابخانه می توانند مدت زمان نگهداری کتابهای عادی و مرجع را حداکثر دو بار تمدید نمایند، اما تمدید مجلات امکان پذیر نمی باشد. میزان تمدید همواره به همان میزان اولیه است؛ یعنی هر استاد با یک بار تمدید کتاب عادی ۱۰ روز دیگر می تواند آن کتاب را نزد خود نگه دارد. تمدید یک مستند باید پیش از پایان زمان مجاز تحویل آن مستند اتفاق بیافتد و امکان تمدید در همان روزی که مستند دریافت شده است وجود ندارد. همچنین توجه داشته باشید که میزان تمدید همواره از موعد تحویل پیشین مورد محاسبه قرار می گیرد و نه از تاریخ تمدید.

در محاسبه مهلت تحویل دقت نمایید، اگر عضوی یک کتاب عادی را مثلاً در روز ۱۳۹۹/۹/۱ امانت گرفته باشد، مهلت تحویل آن ۱۳۹۹/۹/۱۱ خواهد بود. یعنی در صورت تحویل در آن روز جریمه ای به وی تعلق نخواهد گرفت. در صورتی که در روزی کوچکتر یا مساوی ۱۳۹۹/۹/۱۱ اقدام به تمدید نماید، مهلت تحویل جدید ۱۳۹۹/۹/۲۱ خواهد بود.

سابقه اعضای کتابخانه در محاسبه جریمه آنها تاثیرگذار است. به این صورت که تعداد کل روزهایی که فرد در تحویل مستندهای قبلی دیرکرد داشته است، در محاسبه هزینه جریمه های بعدی طبق قواعد زیر تاثیر می گذارد:

|           |  |
|-----------|--|
| کتاب مرجع | برای ۳ روز اول دیرکرد، هزینه جریمه هر روز ۵۰۰۰ تومان و برای روزهای بعد هر روز ۷۰۰۰ تومان (مثلاً به ۵ روز دیرکرد، مبلغ ۲۹۰۰۰ تومان جریمه تعلق می گیرد.) |
| کتاب عادی | برای ۷ روز اول دیرکرد، هزینه جریمه هر روز ۲۰۰۰ تومان، برای روزهای ۸ تا ۲۱، هر روز ۳۰۰۰ تومان و برای روزهای بعد هر روز ۵۰۰۰ تومان                       |
| مجله      | برای مجلات منتشر شده در سال های قبل از ۱۳۹۰، هر روز ۲۰۰۰ تومان و برای مجلات منتشر شده در سال های بعد از آن، هر روز ۳۰۰۰ تومان                          |

## رابط با برنامه زبان جاوا

برنامه‌هایی که برای حل مساله فوق نوشته می‌شوند رابط زیر را پیاده‌سازی می‌نمایند.

```
public interface ILibrary {
    void addStudentMember(String studentId, String studentName);
    void addProfMember(String profName);
    void addBook(String bookTitle, int copies);
    void addMagazine(String magazineTitle, int year, int number, int copies);
    void addReference(String referenceTitle, int copies);
    void borrow(String memberName, String documentTitle);
    List<String> availableTitles();
    void extend(String memberName, String documentTitle);
    void returnDocument(String memberName, String documentTitle);
    int getTotalPenalty(String memberName);
    void timePass(int days);
}
```

شرح هر یک از متدها در جدول زیر ذکر شده است.

|  |   |
|--|---|
| addStudentMember<br>addProfMember      | <p>اضافه کردن عضو دانشجو / استاد</p> <p>خطاها:</p> <ul style="list-style-type: none"> <li>• عضویت فردی با همان نام</li> <li>• تھی بودن رشته‌های نام یا شماره دانشجویی (فرمت شماره دانشجویی مهم نیست)</li> </ul>   |
| addBook<br>addMagazine<br>addReference | <p>اضافه کردن کتاب / مجله / مرجع (اعلام خطا در صورت وجود مستندی با همان عنوان)</p> <p>خطاها:</p> <ul style="list-style-type: none"> <li>• وجود مستندی با همان عنوان (title)</li> <li>• تھی بودن رشته عنوان</li> <li>• صفر یا منفی بودن سال انتشار (year) یا شماره (number) مجلات</li> </ul> |
| borrow                                 | امانت گرفتن مستندی با عنوان documentTitle توسط عضوی با نام memberName   |
| extend                                 | تمدید کردن مستندی با عنوان documentTitle توسط عضوی با نام memberName  |
| returnDocument                         | بازگرداندن مستندی با عنوان documentTitle توسط عضوی با نام memberName  |
| getTotalPenalty                        | مبلغ کل جریمه تعلق گرفته به عضوی با نام memberName، شامل جریمه کتاب‌هایی که تحویل داده و جریمه کتاب‌هایی که اکنون در امانت دارد   |
| availableTitles                        | بررداری از عناوین مستندات در دسترس برای امانت گرفتن   |
| timePass                               | <p>گذشت زمان به مدت days روز (بخش مدل‌سازی زمان را در ادامه متن ببینید)</p> <p>خطاها:</p> <ul style="list-style-type: none"> <li>• منفی بودن days</li> </ul>  |

مثالی از استفاده از این کلاس به این شکل است:

```
private static void testLibrary(ILibrary lib) {
    try {
        lib.addStudentMember("810199999", "Negar");
        lib.addStudentMember("810198888", "Naser");
        lib.addProfMember("Dr. Negargar");
        lib.addProfMember("Dr. Naseri");
        lib.addBook("Lean Startup", 1);
        lib.addMagazine("Today News, vol. 38, no.3", 1395, 3, 1);
        lib.addReference("Cambridge Dictionary", 3);
        lib.borrow("Negar", "Lean Startup");
        System.out.println(lib.availableTitles().size());
        lib.timePass(5);
        lib.extend("Negar", "Lean Startup");
        lib.timePass(50);
        lib.returnDocument("Negar", "Lean Startup");
        System.out.println(lib.getTotalPenalty("Negar"));
    } catch (LibraryException e) {
        System.err.println("Error: " + e.getMessage());
    }
}
```

## مدلسازی زمان

برای اینکه در این پروژه از دروسهای مدیریت تاریخ راحت باشیم، فرض کنید تاریخ را با یک عدد صحیح مدل می‌کنیم. در ابتدای ساخت یک Library، زمان جاری در یک مبدا از پیش تعریف شده قرار دارد و هر بار که متد timePass(d) صدا می‌شود، زمان جاری به مدت d روز به جلو می‌رود. فرض می‌شود تمام متدهای دیگر هنگام فراخوانی در زمان جاری انجام می‌شوند.

## رسیدگی به خطاها

در اجرای هر یک از متدهای کلاس Library، لازم است تمام خطاهای منطقی ممکن کنترل شوند. برای این کار لازم است از مکانیزم رسیدگی به استثناها در جاوا استفاده می‌شود. فرض می‌شود تمام متدهای ILibrary یک استثنا از تایپی سازگار با LibraryException پرتاب می‌کنند که زیر کلاسی از java.lang.exception است. جزئیات استثناها برای ما مهم نیست اما مهم است برای تمام خطاها استثنا پرتاب شود. برخی از خطاها در جدول فوق مشخص شده است. تشخیص بقیه خطاها به شما واگذار شده است. منظور از خطا، هر مشکلی است که منجر شود هدف کاربر از فراخوانی متد محقق نشود. برای مثال، اگر عضوی بخواهد کتابی را تمدید کند اما قبلاً دو بار تمدید کرده باشد، این نیز مصداقی از بروز خطا خواهد بود. در صورت بروز هر خطا، اجرای متد مربوطه روی وضعیت سیستم بدون تاثیر خواهد بود. به عنوان مثال اگر در متد addStudentMember عضوی با نام مشخص شده از قبل موجود باشد، عضو جدید نباید اضافه شود.