



بسمه تعالی



دانشگاه مهندسی برق و کامپیوتر

برنامه‌سازی پیشرفته - پاییز ۱۳۹۹

آزمون عملی میان‌ترم دوم

شرح مسئله

در این مسئله قرار است قواعد امانت‌گیری مستندات یک کتابخانه پیاده‌سازی شود. در این کتابخانه دانشجویان و اساتید می‌توانند انواع مستندات موجود 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 گرفته‌اند و مدت زمان دیرکرد هر کتاب را بازایی نماییم. اعضای کتابخانه می‌توانند علاوه بر کتابهای عادی، کتابهای مرجع و مجلات را نیز از کتابخانه امانت بگیرند. در هر لحظه، یک استاد می‌تواند حداکثر ۵ مستند و یک دانشجو حداکثر ۲ مستند را در امانت داشته باشد. از هر یک از مستندات ممکن است چند نسخه در کتابخانه موجود باشد. در این صورت بیش از یک نفر می‌تواند آن مستند را همزمان به امانت بگیرند. بدیهی است که اگر تمام نسخه‌ها در امانت باشد آن مستند برای امانت‌گیری در دسترس نخواهد بود.

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

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

اعضای کتابخانه می‌توانند مدت زمان نگهداری کتابهای عادی و مرجع را حداکثر دو بار تمدید نمایند، اما تمدید مجلات امکان‌پذیر نمی‌باشد. میزان تمدید همواره به همان میزان اولیه است؛ یعنی هر استاد با یک بار تمدید کتاب عادی ۱۰ روز دیگر می‌تواند 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 می‌گیرد 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 جریمه‌ای 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 تأثیرگذار است. به این صورت که تعداد کل روزهایی که فرد در تحویل مستندهای قبلی دیرکرد داشته است، در محاسبه هزینه جریمه‌های بعدی طبق قواعد زیر تأثیر می‌گذارد:

| | |
|-----------|---|
| 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 می‌گیرد) |
| 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 مجله‌های منتشر شده در سال‌های قبل از ۱۳۹۰، هر روز ۲۰۰۰ تومان و برای مجلات منتشر شده در سال‌های بعد از آن، هر روز ۳۰۰۰ تومان |

واسط کار با برنامه شما

واسط کلی کار با برنامه شما کلاس Library است که دارای رابطی عمومی به این شکل است:

```
class Library {
public:
    void add_student_member(string student_id, string student_name);
    void add_prof_member(string prof_name);
    void add_book(string book_title, int copies);
    void add_magazine(string magazine_title, int year, int number, int copies);
    void add_reference(string reference_title, int copies);
    void borrow(string member_name, string document_title);
    void extend(string member_name, string document_title);
    void return_document(string member_name, string document_title);
    int get_total_penalty(string member_name);
    vector<string> available_titles();
    int time_pass(int days);
private:
    // hidden implementation
};
```

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

| | |
|--|--|
| add_student_member add_prof_member | اضافه کردن عضو دانشجو / استاد خطاها: <ul style="list-style-type: none">• عضویت فردی با همان نام• تهی بودن رشته‌های نام یا شماره دانشجویی (فرمت شماره دانشجویی مهم نیست) |
| add_book add_magazine add_reference | اضافه کردن کتاب / مجله / مرجع (اعلام خطا در صورت وجود مستندی با همان عنوان) خطاها: <ul style="list-style-type: none">• وجود مستندی با همان عنوان (title)• تهی بودن رشته عنوان• صفر یا منفی بودن سال انتشار (year) یا شماره (number) مجلات |
| borrow | امانت گرفتن مستندی با عنوان document_title توسط عضوی با نام member_name |
| extend | امانت گرفتن مستندی با عنوان document_title توسط عضوی با نام member_name |
| return_document | بازگرداندن مستندی با عنوان document_title توسط عضوی با نام member_name |
| get_total_penalty | مبلغ کل جریمه تعلق گرفته به عضوی با نام member_name، شامل جریمه کتاب‌هایی که تحویل داده و جریمه کتاب‌هایی که اکنون در امانت دارد |
| available_titles | برداری از عناوین مستندات در دسترس برای امانت گرفتن |
| time_pass | گذشت زمان به مدت days روز (بخش مدل‌سازی زمان را در ادامه متن ببینید) خطاها: <ul style="list-style-type: none">• منفی بودن days |

مثالی از استفاده از این کلاس به این شکل است:

```
Library ut_lib;
ut_lib.add_student_member("810199999", "Negar");
ut_lib.add_student_member("810198888", "Naser");
ut_lib.add_prof_member("Dr. Nagargar");
ut_lib.add_prof_member("Dr. Naseri");
ut_lib.add_book("Lean Startup", 1);
ut_lib.add_magazine("Today News, vol.38, no.3", 1395, 3, 1);
ut_lib.add_reference("Cambridge Dictionary", 3);
ut_lib.borrow("Naser", "Lean Startup");
vector<string> available_docs = ut_lib.available_titles();
ut_lib.time_pass(10);
ut_lib.extend("Naser", "Lean Startup");
ut_lib.time_pass(12);
ut_lib.return_document("Naser", "Lean Startup");
ut_lib.borrow("Naser", "Cambridge Dictionary");
ut_lib.time_pass(8);
int penalty = ut_lib.get_total_penalty("Naser");
```

مدل سازی زمان

برای این که شما را از دروسهای مدیریت تاریخ راحت کنیم، فرض کنید در ابتدای ساخت یک Library، زمان جاری در یک مبدأ از پیش تعریف شده قرار دارد و هر بار که متد `time_pass(d)` صدا می شود، زمان جاری به مدت `d` روز به جلو می رود. فرض می شود تمام متدهای دیگر هنگام فراخوانی در زمان جاری انجام می شوند.

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 استثناءها¹ در سی پلاس پلاس استفاده کنید.

برخی از خطاها در جدول فوق مشخص شده است. تشخیص بقیه خطاها را به شما واگذار می کنیم. منظور ما 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 مصداق از بروز خطا خواهد بود. در صورت بروز هر خطا، لازم است در **cerr**، عبارتی با این قالب نوشته شود:

Error: <your error message>

مثلاً برای خطای عدم امکان تمدید:

Error: the member has already extended the document twice

تعیین پیغامهای خطا را به سلیقه خودتان انجام دهید. نکته مهم این است که در صورت بروز هر خطا، اجرای متد مربوطه روی وضعیت سیستم بدون تأثیر خواهد بود. به عنوان مثال اگر در متد `add_student_member` عضوی با نام مشخص شده از قبل موجود باشد، عضو جدید نباید اضافه شود. در صورت بروز خطا در یکی از توابع، اجرای برنامه ادامه می یابد و متوقف نمی شود.

کلاس Library خروجی دیگری نباید تولید کند.

¹ exception handling

نحوه تحویل

به منظور سهولت تصحیح، برنامه‌تان را فقط در یک فایل بنویسید که نام فایل شماره دانشجویی شما و پسوند آن cpp است. این فایل را تا پیش از زمان مقرر در صفحه ای‌لرن درس در محل مشخص شده با عنوان میان‌ترم دوم [آپلود] آپلود نمایید.

ارزیابی برنامه

در ارزیابی برنامه شما معیارهای زیر مدنظر قرار می‌گیرند.

| امتیاز | معیارهای ارزیابی برنامه |
|--------|---|
| ۵۰ | درستی عملکرد (شامل پوشش تمام موارد خطا) |
| ۲۰ | طراحی مناسب شیء‌گرا شامل موارد زیر <ul style="list-style-type: none">• استفاده مناسب از وراثت و چندریختی• مخفی‌سازی اطلاعات تا حد ممکن• سپردن مسئولیت محاسبات به کلاس‌های مناسب |
| ۱۰ | تعریف و رسیدگی مناسب به استثناءها |
| ۲۰ | رعایت خوانایی برنامه <ul style="list-style-type: none">• نام‌گذاری مناسب• پرهیز از توابع طولانی• رعایت اصول دندانه‌گذاری^۲• نام‌دهی به‌جا به ثابت‌ها |
| ۱۰۰ | مجموع |

ترتیب پیاده‌سازی توابع

برای این که در صورت مواجهه با کمبود وقت بیشترین امتیاز را از درستی برنامه کسب کنید، توصیه می‌شود پس از طراحی کلاس‌های مرتبط با اعضاء و مستندات، توابع کلاس Library را به این ترتیب پیاده‌سازی کنید:

۱. متدهای `add_*` و متد `time_pass`

۲. متدهای `return_document`، `borrow` (بدون در نظر گرفتن جریمه) و `available_titles`

۳. متد `extend`

۴. متد `get_total_penalty` (و اضافه کردن محاسبات جریمه در بخش‌های مورد نیاز)

^۲ indentation