شاهد لایبرری

وحید که بسیار کثیف کد میزند به دنبال کتابیست که با راهنمایی آن بتواند کد هایش را مرتب بنویسد. استادش به او کتاب clean code اثر رابرت مارتین را معرفی کرده. او که پول کافی برای خرید این کتاب را ندارد به کتابخانه دانشکده مراجعه میکند ولی از بدشانسیش کتابخانه آن کتاب را ندارد. حالا او باید با پای پیاده به کتابخانه تمام دانشکده ها سر بزند. برای جلوگیری از این موضوع، یک سیستم یکپارچه برای کتابخانه های دانشگاه بنویسید که وحید بتواند با یک سرچ سریع و بی دردسر کتاب مورد نظرش را پیدا کند.

برای بهتر درک کردن سوال، یکبار آن را تا انتها بخوانید و سپس شروع به کد زدن کنید.

اینام BookType

در این `enum` انواع کتاب های در دسترس نوشته شده است.

۱. SCIENTIFIC
۲. CRIME
۳. FANTASY
۴. HORROR
۵. CLASSICS

کلاس Publisher

کلاس `Publisher` دارای ویژگی های زیر میباشد. `name` نام ناشر و `location` آدرس ناشر و `id` که شناسه‌ی عددی است که با استفاده از آن می‌توان به آن ناشر دسترسی داشت، اولین ناشری که ساخته می‌شود شناسه‌ی عددی ۱، دومین ناشر، شناسه‌ی عددی ۲ و i-امین ناشر، شناسه‌ی عددی i دارد.

```
int id;
string name;

```

```
string location;
```

کلاس `Publisher` دارای کانستراکتور زیر میباشد.

```
Publisher(string name, string location){
    //TODO
}
```

کلاس Book

کلاس `Book` دارای ویژگی های زیر میباشد. `name` نام کتاب، `type` نوع کتاب، `publisher` ناشر کتاب و ویژگی `borrowed` نشاندهنده موجود بودن کتاب در کتابخانه است. همچنین ویژگی id در این کلاس نیز وجود دارد و مانند کلاس publisher می باشد.

```
int id;
string name;
BookType type;
Publisher publisher;
bool borrowed;
```

کانستراکتور کلاس `Book` به این صورت است.

```
Book(string name, Publisher publisher, BookType type){
    //TODO
}
```

این کلاس یک متود با نام showInfo دارد که ورودی ای ندارد. اگر فرض کنیم نام کتاب `Deep Work` و id آن 4 باشد، خروجی آن یک رشته با فرمت زیر است:

```
4. Deep Work
```

کلاس Member

کلاس `Member` دارای ویژگی های زیر میباشد. `id` کد ملی، `name` نام افراد و `books` کتاب هایی که امانت گرفته اند.

نکته: هر کاربر نباید بیشتر از 5 کتاب امانت بگیرد.

```
string id;
string name;
vector<Book> books;
```

کلاس `Member` دارای کانستراکتور زیر است.

```
Member(string id, string name){
    //TODO
}
```

کلاس Library

کلاس `Library` دارای ویژگی ها زیر است. `id` شناسه‌ی عددی است که با استفاده از آن می‌توان به آن کتابخانه دسترسی داشت، اولین کتابخانه ای که ساخته می‌شود شناسه‌ی عددی ۱، دومین کتابخانه، شناسه‌ی عددی ۲ و i-امین کتابخانه، شناسه‌ی عددی i دارد. `name` نام کتابخانه، `books` آرایه ای از کتاب های موجود در کتابخانه و `position` موقعیت کتابخانه را نشان میدهد.

```
int id;
string name;
vector<Book> books;
int position;
```

کانستراکتور این کلاس به این صورت است.

```
Library(string name, int position){
    //TODO
}
```

این کلاس باید حاوی متدهای زیر باشد:

۱. متدی برای جستجوی یک کتاب با نام آن کتاب
۲. متدی برای برگرداندن لیست کتاب های موجود
۳. متدی برای اضافه کردن کتاب
۴. متدی برای برگرداندن لیست کتاب ها بر اساس نوع کتاب

(در صورت نیاز متد های دیگری نیز تعریف کنید)

کلاس LibrariesHandler

`LibrariesHandler` مهم ترین کلاس این سوال میباشد.

همچنین شامل آرایه ای از کتابخانه های دانشگاه شاهد است. فرض گرفتیم که سیستم یکپارچه است و اگر کسی یک بار ثبتنام کرد میتواند از هر کتابخانه ای که میخواهد کتاب قرض بگیرد.

متدها

- متد createLibrary

این متد یک اسم و یک موقعیت را از ورودی دریافت میکند. اگر کتابخانه ای هم نام با این کتابخانه یا در موقعیت داده شده وجود داشته باشد کتابخانه جدید ساخته نمیشود. در صورتی که نام کتابخانه تکراری باشد یک Exception با متن `A library with this name already exists` و در صورتی که موقعیت تکراری باشد یک Exception با متن `There is now a library in this place` پرتاب (throw) کند.

```
void createLibrary(string name, int position){
    //TODO
}
```

- متد addBook

این متد 2 overload دارد. یکی از آنها آیدی کتابخانه، نام، ناشر و نوع کتاب را دریافت میکند. و کتاب دریافت شده را به کتابخانه داده شده اضافه میکند.

(بر اساس نظر خودتان هر جا اتفاق بدی رخ داد یک Exception با متن مناسب throw کنید)

```
void addBook(int libId, string name, Publisher publisher, BookType type){
    //TODO
}
```

متد دوم که همین نام را دارد ورودی اش یک آیدی کتابخانه و یک شی از کلاس Book است.

```
void addBook(int libId, Book book){
    //TODO
}
```

- متد addMember

این متد نام و کدملی یک شخص را دریافت میکند. اگر شخصی با این کد ملی قبلا در سیستم وجود داشت ثبتنام انجام نمیشود و یک Exception پرتاب میکند.

```
void addMember(string name, string id){
  // TODO
}
```

- متد getAllBooks

این متد از ورودی آیدی یک کتابخانه را دریافت میکند و تمام کتاب های موجود در آن را به ترتیبی که اضافه کرده بودیم برمیگرداند.

نکته: اگر کتابخانه ای با این آیدی وجود نداشت، [] برگردانید.

امضای این متد به این صورت است.

```
vector<Book> getAllBooks(int libId){
    // TODO
}
```

- متد getAllBooksInfo

این متد از ورودی آیدی یک کتابخانه را دریافت میکند و اطلاعات تمام کتاب های موجود در آن را به ترتیبی که اضافه کرده بودیم به صورت زیر برمیگرداند. برای مثال اگر ابتدا کتاب `clean code` و سپس کتاب `the old man and see` و سپس کتاب `cpp` به لیست کتاب های کتابخانه با آیدی 1 اضافه شوند، خروجی متد به این صورت است.

```
1. clean code
2. the old man and see
3. cpp
```

نکته: اگر کتابخانه ای با این آیدی وجود نداشت، یک رشته خالی برگردانید.

امضای این متد به این صورت است.

```
string getAllBooksInfo(int libId){
    // TODO
}
```

- متد filterByType

این متد آیدی یک کتابخانه و نوع کتاب را از ورودی دریافت میکند و کتاب هایی که در این کتابخانه موجودند را بر اساس تایپ داده شده فیلتر میکند و لیست فیلتر شده را بر میگرداند.

امضای این تابع به این صورت است.

```
vector<Book> filterByType(int libId, BookType type){
    // TODO
}
```

در وصورت وجود نداشتن کتابخانه ای با آیدی داده شده، [] برگردانید.

- متد filterByTypeAndShowInfo

این متد آیدی یک کتابخانه و نوع کتاب را از ورودی دریافت میکند و کتاب هایی که در این کتابخانه موجودند را بر اساس تایپ داده شده فیلتر میکند و لیست فیلتر شده را به فرمت زیر بر میگرداند.

```
1. clean code
2. cpp
3. python object oriented programming
```

امضای این تابع به این صورت است.

```
string filterByTypeAndShowInfo(int libId, BookType type){
    // TODO
}
```

در وصورت وجود نداشتن کتابخانه ای با آیدی داده شده، رشته خالی برگردانید.

- متد borrow

این متد آیدی شخص و کتابخانه و نام یک کتاب را دریافت میکند و پس از بررسی در صورت عدم مشکل، کتاب قرض داده میشود در غیر اینصورت کاری انجام نمیشود.

```
bool borrow(string memberId, int libraryId, string name){
    //TODO
}
```

- متد returnBook

این متد نیز همانند متد borrow، آیدی شخص و کتابخانه و نام یک کتاب را دریافت میکند. و پس از بررسی در صورت عدم مشکل، کتاب به کتابخانه برگردانده میشود در غیر اینصورت کاری انجام نمیشود

```
bool returnBook(string memberId, int libraryId, string name){
    //TODO
}
```

- متد size

این متد تعداد کتابخانه های موجود در سامانه را برمیگرداند.

```
public int size(){
    // TODO
}
```

- متد findNearestLibraryByPosition

این متد نام یک کتاب و مختصات فعلی ما را دریافت میکند و نزدیک ترین کتابخانه ای را که این کتاب در آن موجود است به ما برمیگرداند.

اگر این کتاب در هیچ کتابخانه ای وجود نداشت، عدد -۱ برگردانده شود

نکته: اگر فاصله دو کتابخانه از ما برابر بود باید نام کتابخانه ای که آیدی کوچکتری دارد برگردانده شود.

```
Library findNearestLibraryByPosition(string name, int position){
    // TODO

```

```
}
```

- متد findLibrariesHaveBook

این متد نام یک کتاب و مختصات فعلی ما را دریافت میکند و یک لیست از کتابخانه هایی که این کتاب در آن موجود است به فرمت زیر برگرداند. **لیست خروجی باید بر اساس فاصله ما تا کتابخانه از کمترین فاصله تا بیشترین فاصله مرتب شده باشد.**

ابتدا شماره لیست، سپس نام کتابخانه و در اخر فاصله ما تا کتابخانه نمایش داده میشود.

```
1. Markazi 1
2. Computer 3
3. khabgah 20
```

امضای این متد به این صورت است.

```
string findLibrariesHaveBook(string name, int position){
    //TODO
}
```

نکات

- همه فیلدها یا attributeها private باشند و در صورت نیاز برای آنها setter و getter بنویسید. (فیلدهای static را می توانید public بگذارید)
- در صورت نیاز میتوانید متدها و فیلدهای جدیدی به کلاس ها اضافه کنید.
- تضمین میشود هر کتابخانه تنها یک نسخه از هر کتاب را داشته باشد.
- هر جا نیاز بود و در متن تمرین گفته نشده بود، Exception پرتاب کنید.

نحوه ارائه تمرین

- یک ریپازیتوری **Private** با نام AP-HW2-ShahedLibrary در گیت هاب خود بسازید و آن را در لوکال خود clone کنید. حال فایل یا فایل های تمرین خود را در این ریپازیتوری بسازید و پس از هر تغییر، commit کنید.
- پس از اتمام تمرین تان اکانت گیت هاب ما (**Shahed-University-AP-TA-1401**) رو به عنوان collaborator اد کنید و آدرس ریپازیتوری را در یک فایل txt قرار دهید و آپلود کنید.
- تحویل تمرین تان بعد از اتمام زمان آن و با حضور شما و یکی از TAها انجام می شود.