

## به نام خدا

### سامانه مدیریت کتابخانه

در این تمرین سعی داریم تا یک سامانه مدیریت کتابخانه را مدل کنیم. این کتابخانه قرار است یک سری کتاب را در کتابخانه نگهداری کند و به اعضای کتابخانه امانت دهد. اعضا می توانند کتاب را از کتابخانه به امانت گیرند و یا آن را پس دهند. در طراحی این سامانه، شما باید مدیریت اعضای کتابخانه، را انجام دهید و اعمالی از قبیل اضافه کردن کتاب به کتابخانه، حذف کتاب از کتابخانه، به امانت دادن کتاب به اعضا و ... در این سامانه قابل انجام باشد.

نکته: ممکن است برای پیاده سازی کلاس های خود نیازمند اضافه کردن توابع `private` یا `public` به کلاس ها باشید. توابع زیر صرفا توابعی هستند که باید پیاده سازی شوند. همچنین در موارد مورد نیاز متدهای `get` و `set` برای متغیرها بنویسید.

برای مدلسازی مساله بالا نیازمند کلاس های زیر هستیم.

```
public class Library{
```

این کلاس برای مدلسازی کتابخانه بصورت کلی استفاده می شود.

```
public class Member{}
```

این کلاس برای مدلسازی اعضای کتابخانه استفاده می شود.

```
public class Borrow{}
```

این کلاس برای مدلسازی به امانت گرفتن کتاب استفاده می شود.

```
public class Book{}
```

این کلاس برای توصیف کتاب هایی که در کتابخانه به امانت داده می شوند، استفاده می شود..

```
public class Writer{}
```

این کلاس برای مدلسازی نویسندگان استفاده می شود.

```
public class Address{}
```

این کلاس برای مدلسازی آدرس یک شخص استفاده می شود.

متدهای کلاس های زیر بدین صورت هستند:

Library: شرح متدهای کلاس:

```
public Library(String name);
```

سازنده: اسم کتابخانه را به عنوان ورودی دریافت می کند.

```
public void addMember(Member m);
```

این تابع یک عضو به لیست اعضای کتابخانه اضافه می کند.

```
public Member[] getMembers();
```

لیست اعضای کتابخانه را بازمی گرداند..

```
public void addBook(Book b);
```

یک کتاب جدید را به لیست کتاب های کتابخانه اضافه می کند.

```
public Book[] getExistBooks();
```

یک آرایه از کتاب های موجود در کتابخانه را باز می گرداند.

```
public Book[] getBorrowedBooks();
```

یک آرایه از کتاب های قرض داده شده بر می گرداند.

```
public Book[] getAllBooks();
```

یک آرایه از کلیه کتاب‌هایی که کتابخانه دارد ( چه امانت گرفته شده و چه موجود) برمی‌گرداند.

```
public Borrow[] getBorrows();
```

یک آرایه از تمام امانت‌های بر می‌گرداند.

```
public Borrow getBorrow(int id);
```

امانتی با شناسه id را برمی‌گرداند.

متدهای کلاس: **Book**

```
public Book(String name, int ID, Writer[]writers, int year);
```

سازنده: اسم کتاب، کد کتاب، نویسندگان کتاب و سال نشر کتاب را به عنوان ورودی می‌گیرد.

برای هر یک از ویژگی‌های این کلاس متد **set** و **get** نوشته شود.

متدهای کلاس: **Member**

```
public Member(String name, int ID, Address address);
```

سازنده: نام عضو کتابخانه و شناسه و آدرس آن را می‌گیرد.

```
public Member(String name, int ID);
```

سازنده: نام عضو کتابخانه و شناسه آن را می‌گیرد.

```
public int getNumBorrows();
```

تعداد کتاب‌های کنونی به امانت گرفته شده توسط یک عضو را باز می‌گرداند.

```
public void addBorrow(Borrow borrow);
```

یک امانت را به لیست امانت‌های عضو اضافه می‌کند.

```
public Borrow[] getTotalBorrows();
```

تمام درخواست‌های به امانت گرفتن کتاب را باز می‌گرداند.

```
public Member getMember(int id);
```

عضوی با شماره شناسه id را برمی‌گرداند. در صورت عدم وجود عضوی با این شماره شناسه null بر می‌گرداند.

متدهای کلاس: **Borrow**

```
public Borrow(int ID, Member m);
```

سازنده: شناسه درخواست امانت و عضو درخواست دهنده را دریافت می‌کند.

```
public int getID();
```

شناسه خرید را باز می‌گرداند.

```
public void addItem (Book book);
```

کتاب **book** را به لیست امانت اضافه می‌کند.

```
public void removeItem (Book book);
```

کتاب **book** را از لیست امانت حذف می‌کند.

```
public Book[] getItems();
```

خروجی این تابع یک آرایه از کتاب‌های موجود در درخواست امانت است.

متدهای کلاس: **Writer**

```
public Writer(String firstName, String lastName);
```

سازنده: نام و نام خانوادگی نویسنده را دریافت می‌کند.

متدهای کلاس: **Address**

```
public Address(String provinceName, String townName, String streetName, int plaque);
```

سازنده: نام استان، نام شهر، نام خیابان و شماره پلاک را دریافت می‌کند.

نحوه‌ی ارتباط این سامانه با کاربرانش با دستورات ورودی در **terminal** است و برنامه شما باید در یک حلقه بی‌نهایت ورودی بخواند. قالب دستورات این سامانه بصورت سلسله‌مراتبی است. بدین صورت که در ابتدا نوع دستور **add , remove, report** مشخص می‌شود. سپس موجودیتی که قرار است این دستور روی آن اعمال شود **member , book , borrow** . در نهایت نیز اطلاعات موجودیت وارد می‌شود.

برنامه با دستور **terminate** خاتمه می‌یابد.

**دستور: add**

در صورتی که این دستور وارد شد در خط بعدی کاربر موجودیت مورد نیاز خود را وارد می‌کند.

**member**

در صورتی که موجودیت مشتری بود در خط‌های بعدی کاربر مشخصات مشتری را بدین صورت وارد می‌کند.

شناسه مشتری

نام مشتری

آدرس مشتری به ترتیب: نام استان، نام شهر، نام خیابان، پلاک، هر کدام در یک خط.

**book**

در صورتی که موجودیت کتاب بود در خط‌های بعدی کاربر مشخصات کتاب را بدین صورت وارد می‌کند.

شناسه کتاب

نام کتاب

تعداد نویسندگان

مشخصات نویسندگان کتاب به ترتیب: نام، نام خانوادگی هر کدام در یک خط

سال نشر کتاب

**borrow**

شناسه درخواست امانت

شناسه کاربر درخواست دهنده

**item**

این دستور برای اضافه کردن کتاب به درخواست امانت عضو است. در خط‌های بعدی مشخصات **item** بصورت زیر وارد می‌شود.

شناسه درخواست امانت

شناسه کتاب

پس از این دستور در صورتی که کتاب در کتابخانه موجود بود در امانت کسی نبود به لیست امانت مشخص شده اضافه می‌شود همچنین هر یک از آرایه‌های **borrowedBooks** و **existBooks** در کلاس **Library** آپدیت می‌شوند. ولی در صورتی که موجود نبود اضافه نمی‌شود.

**دستور: report**

**customers**

در این حالت باید اطلاعات اعضای کتابخانه هر کدام در یک خط پرینت شود. فرمت خروجی هر مشتری بدین صورت است.) از آنجا که خروجی شما به زبان انگلیسی هست فرمتهای خروجی نیز به زبان انگلیسی هست.)

member-ID, member-name, total-borrow-size, member address

دقت کنید آدرس مشتری به ترتیب نام استان، نام شهر، نام خیابان، پلاک پرینت شود.

books

در این حالت تعداد کتابهای موجود در کتابخانه و قابل به امانت گرفتن پرینت می شود.

borrow

در این حالت شناسه درخواست امانت در یک خط به عنوان ورودی گرفته می شود و پس از آن در خط دیگر مشخصات آن امانت به ترتیب زیر در خروجی نشان داده می شود:

borrow\_ID, member\_ID, booksList

که لیست کتابها به ترتیب به صورت زیر نمایش داده می شود :

book\_ID, book\_name, book\_year, book\_writer1\_name, book\_writer2-name , ...

دستور: remove

item

این دستور یک کتاب را از لیست امانت عضو حذف می کند. ورودی خطهای بعدی برای توصیف این دستور بدین صورت است.

شناسه درخواست

شناسه کتاب

## نمونه ورودی :

add

member

1

ali

fars

shiraz

jamejam

24

report

customers

terminate

نمونه خروجی:

1,ali,0,fars,shiraz,jamejam,24