به نام خدا

تمرین دوم درس برنامه‌نویسی پیشرفته

۰. فایل مربوط به توضیحات نحوه ارسال تمرین‌ها را که در مودل قرار دارد، مطالعه کنید.

۱. تمامی فایل‌های کد را به همراه فایل متنی که در قالب pdf است (مورد سوم را بخوانید) به صورت یک فایل آرشیو zip (zip != rar) که به قالب زیر نام‌گذاری شده است، بارگذاری نمایید.

StudentNumber_FirstName_LastName.zip

9031066_Ehsan_Edalat.zip

۲. در سوال‌هایی که ورودی و خروجی مطلوب آن‌ها مشخص شده است، برنامه‌ی شما به صورت ماشینی تصحیح می‌شود. بنابراین رعایت نحوه ورودی‌گرفتن و نمایش خروجی اهمیت بسیاری دارد. دقیقا همان‌طور که از شما خواسته شده است ورودی‌ها را خوانده و خروجی‌ها را تولید کنید.

۳. پاسخ سوالات تشریحی را به صورت تایپ‌شده و در قالب **یک فایل pdf** (برای کل تمرین) تحویل دهید.

مهلت تحویل: تا جمعه ۱۶ اسفند ۱۳۹۸ ساعت ۲۳:۵۵ شب

سوال اول

تعریف مفاهیم:

Class چیست؟ اجزای تشکیل دهنده یک کلاس را نام ببرید.

تفاوت Class method با یک Function عادی در چیست؟

کنترل دسترسی(Access Control) به ویژگی برای (attribute یا field یا property) های یک کلاس چگونه است؟ در متد ها چطور؟

یک مثال بیاورید از حالتی که تغییر مقدار یک متغیر به طور مستقیم و بدون استفاده از setter، مشکل ایجاد کند.

سوال دوم

درست یا غلط بودن موارد زیر را با توضیح مختصری تعیین کنید.

۱. اعضای protected در یک کلاس تنها برای اعضای همان کلاس قابل دسترسی هستند.
۲. اگر در روند برنامه از کلاسی یک شیء ایجاد کنیم، تنها در زمان اجرای برنامه است که به آن شیء، حافظه اختصاص می‌یابد و پیش از آن در حافظه وجود ندارد.
۳. فرض کنید کلاسی public با نام A تعریف کردیم. این کلاس حاوی یک متد public با نام b می‌باشد. این متد از طریق A.b() قابل دسترسی است.
۴. درصورتی که ما یک constructor با چند پارامتر تعریف کنیم زبان جاوا نیز یک default constructor بدون argument به صورت پیش فرض در نظر می‌گیرد.
۵. Type مقادیر اعشاری موجود در کد به صورت پیشفرض float است.
۶. جاوا یک زبان برنامه‌نویسی statically typed است به این معنا که پیش از استفاده از هر متغیر باید حتما آن را با Type مشخص تعریف نمود.
۷. در تعریف توابع return type پیش از visibility modifier نوشته می‌شود.

سوال سوم

در جاوا چندین راه برای پاس‌دادن تعداد نامعینی از متغیرها به یک تابع وجود دارد، یکی از این راه‌ها ارسال متغیرها به صورت یک آرایه (یا ArrayList یا List و یا کلاس‌های دیگر شناخته شده) است. یکی دیگر از این راه‌ها استفاده از Syntaxای با نام Varargs (= Variable Arguments) است که به این شکل در زبان پیاده می‌شود.

```
class Test1
{
    // A method that takes variable number of integer
    // arguments.
*   static void fun(int... a)
    {
        System.out.println("Number of arguments: " + a.length);
        // using for each loop to display contents of a
**      for (int i: a)
                System.out.print(i + " ");
        System.out.println();
    }

    // Driver code
    public static void main(String args[])
    {
        // Calling the varargs method with different number
        // of parameters
        fun(100);             // one parameter
        fun(1, 2, 3, 4); // four parameters
        fun();          // no parameter
    }
}
```

در خطی که با * مشخص شده، می‌بینیم که با استفاده از ساختار type_name...، تابع پارامتری با نام a را به صورت مجموعه (Collection)ای از تایپ مشخص شده دریافت می‌کند.

در خطی که با ** مشخص شده، می‌بینیم که با استفاده از ساختار:

type_name name: collection

یک for-each روی متغیرهای مجموعه‌ی a که عنصر فعلی آن با i نشان داده شده و از نوع type است انجام می‌گیرد.

حال، در سوال زیر از شما می‌خواهیم که با استفاده از دانشی که تا به این‌جای کار کسب کرده‌اید، مشکلات کد ضمیمه‌شده در فایل SortArray.java را برطرف کرده و سپس نحوه‌ی عملکرد کد را توضیح دهید.

سوال چهارم

طراحی و مدل سازی یک سیستم جامع مدیریت کتابخانه

در این سوال شما باید طراحی و پیاده سازی یک سیستم مدیریت کتابخانه‌ها را بر عهده بگیرید. در ادامه سوال به طور کامل ویژگی های این سیستم برای شما شرح داده می‌شود.

۱. سامانه کلی مدیریت کتابخانه‌ها (LibrarySystem)

۱ - ۱ . این کلاس لیستی از کتابخانه‌ها دارد (libraries)

این کلاس لیستی از کتابخانه‌ها دارد.

در این کلاس ۳ متد وجود دارد :

1. addLibrary(Library libraryToAdd)
2. removeLibrary(Library libraryToRemove)
3. printAllLibraries()

متدهای بالا به ترتیب :

یک کتابخانه می‌گیرد و آن را به سیستم اضافه می‌کند.

یک کتابخانه می‌گیرد و آن را از سیستم حذف می‌کند.

تمام کتابخانه‌های موجود در سیستم را چاپ می‌کند.

۲. کتابخانه (Library)

۲ - ۱. هر کتابخانه لیستی از کتاب‌ها دارد (books)

۲ - ۲. هر کتابخانه لیستی از کاربران ثبت‌نام شده در سیستم خودش دارد (users)

۲ - ۳. هر کتابخانه یک آدرس (address) و یک نام دارد (name)

۲ - ۴. هر کتابخانه لیستی از قرض‌ها (borrows) دارد.

مقادیر فیلد های address و name کتابخانه در constructor مقدار دهی می‌شوند.

این کلاس دارای متدهای زیر است:

1. addUser(User userToAdd)
2. removeUser(User userToRemove)
3. addBook(Book bookToAdd)
4. removeBook(Book bookToRemove)
5. borrowBook(Book bookToBorrow, User borrower, Date deadline)
6. giveBackBook(Borrow borrow)
7. printPassedDeadlineBorrows()

متدهای اضافه کردن و حذف کردن کاربران و کتاب‌ها از سیستم که نیازی به توضیح ندارد.

دو متد بعدی برای قرض گرفتن و پس دادن کتاب است:

متد borrowBook یک کتاب و یک کاربر و یک تاریخ به عنوان ورودی می‌گیرد و یک instance از کلاس Borrow با ورودی های مربوطه می‌سازد و به لیست قرض های کتابخانه (borrows) اضافه می‌کند.

(طبیعتا کتابی که قرض داده می‌شود از لیست کتاب‌های موجود در کتابخانه حذف می شود.)

متد برگرداندن کتاب نیز یک قرض به عنوان ورودی می‌گیرد و آن قرض را بررسی می کند و عملیات لازم را انجام می‌دهد.

متد printPassedDeadlineBorrows باید از بین قرض‌ها، آن قرض‌هایی که تاریخ برگرداندن کتابشان گذشته است، چاپ کند. خروجی چاپ باید دقیقا به ازای هر قرض، خروجی متد print آن قرض باشد.

۳. کتاب (Book)

۳ - ۱. هر کتاب دارای عنوان است. (title)

۳ - ۲. هر کتاب دارای یک نام نویسنده است. (author)

مقادیر فیلدهای این کلاس نیز توسط constructor مقدار دهی می‌شوند.

این کلاس نیز دقیقا معادل کلاس User دارای دو متد است. یکی متد print و یکی برای چک کردن یکسان بودن ۲ کتاب (از طریق چک کردن نویسنده و عنوان کتاب). خروجی متد print باید دقیقا معادل زیر باشد:

Title: title | Author: author

۴. کاربر (User)

۴ - ۱. هر کاربر دارای یک نام است. (firstName)

۴ - ۲. هر کاربر دارای یک نام خانوادگی است. (lastName)

۴ - ۳. هر کاربر دارای یک شماره ملی است که برای شناسایی کاربر به کار می رود (idNum)

تمامی مقادیر تعریف شده در فیلد ها توسط Constructor مقداردهی می‌شوند.

علاوه بر این تمامی مقادیر در صورت نیاز متدهای getter و setter دارند.

فیلد شماره ملی در این کلاس باید در قالب یک رشته ۱۰ رقمی باشد و یک کتابخانه نباید کاربری با دو شماره ملی یکسان داشته باشد.

این کلاس دو متد نیز دارد. یک متد print برای چاپ کردن اطلاعات کاربر و متد دیگر معادل override شده‌ی متد equals است که یک کاربر می‌گیرد و از طریق مقایسه شماره ملی چک می‌کند که این دو کاربر یکسان هستند یا خیر. خروجی متد print باید دقیقا معادل زیر باشد: (در مورد مفهوم override کردن متدها در آینده به صورت کامل آشنا خواهید شد. پیاده سازی این متد با تکنیک override کردن امتیازی خواهد بود.)

Full Name: firstName lastName | ID: idNumber

۵. قرض (Borrow)

۵ - ۱. هر کلاس قرض یک فیلد به اسم borrower از نوع User دارد که نشان دهنده کاربری است که کتاب را قرض گرفته است.

۵ - ۲. هر کلاس قرض یک فیلد به اسم book از نوع Book دارد که نشان دهنده کتابی است که قرض گرفته است.

۵ - ۳. هر کلاس قرض یک تاریخ دارد (از نوع کلاس Date) به نام issuedDate که نشان دهنده تاریخ شروع قرض است. برای نگهداری زمان تنها روز، ماه، سال و ساعت کافی است.

۵ - ۴ .هر کلاس قرض یک تاریخ دارد (از نوع کلاس Date) به نام deadlineDate که نشان دهنده تاریخ پایان و ددلاین بازگرداندن کتاب است.

برای مطالعه در مورد کلاس Date در جاوا می‌توانید از منبع زیر (و یا هر منبع دیگری) استفاده کنید.

https://www.tutorialspoint.com/java/java_date_time.htm

تمامی مقادیر فیلدهای borrower و deadlinedDate و book و issuedDate باید در constructor مقداردهی شوند. مقدار issuedDate باید تاریخ لحظه ای سیستم در زمان ایجاد قرض باشد.

تمامی متدهای getter و setter این کلاس را به طور کامل پیاده سازی کنید.

بدیهی است که تاریخ deadlineDate نباید قبل از تاریخ issuedDate باشد.

این کلاس یک متد print نیز دارد که اطلاعات مربوط به این قرض را چاپ می‌کند.

خروجی متد print باید دقیقا به صورت زیر باشد:

```
Borrower => Full Name: firstName lastName | ID: idNumber
Book => Title: title | Author: author
IssuedDate => issuedDate
Deadline => deadlineDate
Remaining => deadlineDate - issuedDate
```

توجه شود موارد و حالتهای استثنا یا خاص که برای نمونه تعدادی از آنها در زیر آمده استباید مدیریت شوند و در صورت لزوم پیام مناسب به کاربر داده شود:

- حذف کاربری که کتابی گرفته که هنوز پس نداده.
- حذف کتابی که در سیستم وجود ندارد.
- حذف کتابی که قبلا توسط کسی قرض گرفته شده و پس داده نشده.
- اضافه کردن کاربر (یا کتاب) تکراری به سیستم کتابخانه.
- قرض کتابی که قرض گرفته شده.
- پس دادن کتابی که قرض گرفته نشده.
- پس دادن کتابی که از کتابخانه x گرفته شده به کتابخانهy

انجام دهید:

۱. کلاس دیاگرام تمرین را به طور کامل طراحی کنید.

۲. کلاس‌های بالا را به صورت کامل پیاده‌سازی کنید. (بررسی کردن تمام نکات گفته شده در بالا ضروری است)

۳. یک کلاس برای تست به عنوان کلاس TestLibrarySystem در نظر بگیرید. در این کلاس یک سیستم کتابخانه در نظر بگیرید. در این کلاس ۲ کتابخانه و برای هر کتابخانه ۵ کاربر و ۵ کتاب با عناوین مختلف در نظر بگیرید و تمامی متدهای این چند کلاس را به طور کامل تست کنید.

تمامی حالات استثنا باید در کد تست شما موجود باشد.

کدهای شما باید بدون هیچ خطایی build و run شوند و در صورت عدم اجرای کد نمره ای به شما تعلق نمی‌گیرد.

بدیهی است در این تمرین باید از Collectionهای موجود در جاوا استفاده کنید.

برای مطالعه در این زمینه میتوانید از منبع زیر (یا هر منبع دیگری) استفاده کنید:

https://www.tutorialspoint.com/java/java_collections.htm

اسامی استفاده شده در این تمرین باید عینا اسامی استفاده شده در کد های شما باشد و باید دقیقا با همان فرمت گفته شده کد نوشته شود. جاواداک و رعایت خوانایی کد و کامنت های مفید در قسمت های مختلف کد (انگلیسی!) اجباری است و بخش مهمی از نمره است.