

به نام خدا

تمرین سوم درس برنامه‌نویسی پیشرفته

۰. فایل مربوط به توضیحات نحوه ارسال تمرین‌ها را که در مدل 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. دقیقاً 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 ۸ فروردین ۱۳۹۹ ساعت ۲۳:۵۵ شب

سوال اول

درستی یا نادرستی عبارات زیر را مشخص کنید.

۱. هنگام پیمایش لیست توسط iterator یا for-each، می‌توان عناصر آن لیست را پاک کرد ولی در حلقه‌های while و for با خطا مواجه می‌شویم.
۲. اگر یک شیء را بدون نام (anonymous object) تعریف کنیم و به ArrayList اضافه کنیم در heap ذخیره می‌شود.
۳. یک فیلد final را باید مقداردهی کنیم وگرنه با خطای کامپایل روبه‌رو می‌شویم.

جاهای خالی را با عبارات مناسب پر کنید.

۱. HashSet و HashMap نمی‌توانند داده با Type های را ذخیره کنند.
۲. نوع خاصی از کالکشن است که می‌تواند تنها تعداد ثابتی از عناصر را ذخیره کند.

۳. اگر یک شیء را بدون نام تعریف کنیم و مستقیم به متدی که از آن استفاده می کند پاس بدهیم، یک شیء تعریف کرده ایم.
۴. به ازای هر شیء جدیدی که ساخته می شود، در heap به اندازه ی جمع اندازه تمام فیلدهای آن شیء به علاوه فضا اشغال می شود.
۵. اگر یک بلوک در کلاسی داشته باشیم اولین باری که از آن کلاس یک شیء می سازیم یا اولین باری که به یک عضو static از آن کلاس دسترسی پیدا می کنیم، آن بلوک اجرا می شود. (امتیازی)

سوال دوم

به سوالات زیر پاسخ کامل دهید.

۱. تفاوت بین ArrayList و LinkedList در جاوا چیست؟
۲. فرض کنید به تعداد 1000 عدد داده داریم که هر داده شامل نام و نام خانوادگی و سن و شماره تماس و شماره ملی است. قرار است این داده ها را در یکی از کالکشن های ArrayList و یا LinkedList نگهداری کنیم. می دانیم درصد استفاده و پیدا کردن داده ها خیلی بیشتر از اعمال تغییر در کالکشن ها است (از نظر کم و زیاد کردن). به نظر شما بهتر است از کدامیک از کالکشن های گفته شده استفاده کنیم؟ دلیل خود را توضیح دهید.
۳. کاربرد Iterable (که یک interface در جاوا است) را توضیح دهید. (امتیازی)

سوال سوم

برنامه زیر یک کد InsertionSort است. خطاهای برنامه را پیدا کرده و آن ها را اصلاح کنید تا به درستی آرایه زیر را مرتب کند.

```

public class Main {

    public static void main(String[] args) {
        int[] array = {14, 19, 8, 7, 48, 16, 63, 27};

        int n = array.length;
        for (int j = 1; j < n; j++) {
            int key = array[j];
            int i = j - 1;
            while ((j > 0) && (array[i] > key)) {
                array[i + 1] = array[i];
                i--;
            }
            array[i + 1] = key;
        }
        for (int i : array) {
            System.out.print(i + " ");
        }
    }
}

```

سوال چهارم

در این سوال می‌خواهیم سیستم مدیریت پورتال دانشجویان را شبیه سازی کنیم. دانشجویان دانشگاه هرکدام مشخصاتی دارند و هریک می‌توانند تعدادی درس را هر ترم اخذ نمایند. تعداد واحدهای اخذ شده آن‌ها به معدل کل آن‌ها بستگی دارد و به این صورت است که دانشجویان با معدل بالاتر یا مساوی ۱۵ قادر به اخذ حداکثر ۲۰ واحد، با معدل بیشتر از ۱۰ و کمتر از ۱۵ قادر به اخذ حداکثر ۱۵ و دانشجویانی که معدل کمتر از ۱۰ دارند قادر به اخذ هیچ واحد درسی نمی‌باشند. هر واحد درسی ظرفیتی دارد که براساس اولویت زمانی به دانشجویانی که زودتر درس را اخذ نموده‌اند، اختصاص می‌یابد.

هریک از دانشجویان دارای مشخصات نام، شماره دانشجویی، معدل کل و لیستی از واحدهایی که اخذ کرده‌اند، می‌باشند.

هر واحد درسی نیز دارای مشخصات اسم درس، کد درس، تعداد واحدهای درس، حداکثر ظرفیت و لیستی از دانشجویانی که درس را اخذ کردند می‌باشد.

کلاس دانشجو (student)

- شماره دانشجویی هر فرد یک عدد دقیقاً ۷ رقمی است که در غیر اینصورت فرد جزو دانشجویان دانشگاه شناخته نمی‌شود. مانند شماره دانشجویی خودتان (برای مثال ۹۰۳۱۰۶۶) دو رقم اول سال ورود، دو رقم دوم کد دانشکده و بقیه شماره فرد است.

- معدل کل باید عددی بین ۰ تا ۲۰ باشد.
- هر دانشجو با توجه به محدوده معدل کل خود حداکثر قادر به اخذ تعدادی مجاز از واحدهای درسی است. همچنین پس از اخذ هر واحد درسی می‌تواند واحد را حذف نماید. ولی هیچ‌گاه نمی‌تواند از بازه مجاز تعداد واحد خود عبور نماید.
- یک دانشجو نمی‌تواند یک درس را چند بار اخذ نماید.
- باید متدهایی برای نمایش اطلاعات دانشجو مثل اطلاعات فردی، دروس اخذشده و غیره تعریف شوند. در این متدها با توجه به کارکرد، رشته‌ای شامل اطلاعات مورد نظر باید تولید شوند. برای مثال در نمایش اطلاعات فردی، رشته تولیدشده باید شامل نام، شماره دانشجویی و غیره باشد. با توجه به تعریف پروژه توابع لازم برای نمایش اطلاعات را شناسایی و پیاده‌سازی کنید.

کلاس درس (course)

- کد هر درس عددی ۴ رقمی است (برای مثال ۳۱۰۱). دو رقم اول کد دانشکده و دو رقم دوم شماره درس هستند.
- تعداد واحدهای هر درس حداقل ۱ و حداکثر ۳ می‌باشند.
- تعداد دانشجویانی که قادر به اخذ درس هستند نباید از ظرفیت حداکثر درس عبور کند. در صورت اتمام ظرفیت یک درس، دانشجویان قادر به اخذ آن درس نمی‌باشند.
- هیچ دانشجویی نمی‌تواند یک درس را چند بار اخذ کند، پس لیست دانشجویان کلاس نمی‌تواند عضو تکراری بپذیرد. بدیهی است در صورتی که دانشجویی درسی را حذف نمود، از لیست دانشجویان آن درس حذف می‌گردد.
- برای نمایش اطلاعات هر درس باید متدهایی در نظر گرفته شود.

ورودی‌ها:

ورودی دادن به و خروجی گرفتن از این برنامه از طریق Terminal انجام می‌شود. در ورودی این سوال ابتدا تعداد درس‌ها و سپس اطلاعات هر درس به ترتیب اسم درس، کد درس، تعداد واحدهای درس و بیشینه ظرفیت وارد می‌شود. سپس تعداد دانشجویان و بعد از آن مشخصات هر دانشجو به ترتیب نام، شماره دانشجویی و معدل کل و تعداد درس‌هایی که قصد اخذ آن‌ها را دارد وارد می‌شود. بعد از آن به تعداد دروس درخواستی می‌تواند برای اخذ کد درس و یا نام درس را وارد کند. (در این مرحله باید چک شود که اگر دانشجویی متعلق به این دانشکده نیست نباید بتواند درس‌های ارائه‌شده در آن دانشکده را اخذ نماید.) در نهایت تعداد دانشجویانی که قصد حذف و یا اضافه کردن درسی را دارند ورودی داده می‌شود.

در خطوط بعدی نام دانشجو و تعداد دروس نیازمند تغییر نوشته می‌شود. بعد از آن کد درس‌های مورد نظر به عنوان ورودی خواهند بود. (اگر درس را قبلاً اخذ کرده بود آن را حذف کرده، در غیر این صورت آن را به دروس وی اضافه کنید)

خروجی‌ها:

در خروجی باید لیست اسامی دانشجویان هر کلاس و بعد از آن‌ها اسامی درس‌های اخذ شده توسط هر دانشجو به صورت مجزا چاپ شوند. (ترتیب نمایش هر کدام به ترتیب ورودی هاست).
توجه داشته باشید ورودی و خروجی‌ها دقیقاً مطابق نمونه داده شده باشد و از چاپ پیام‌های اضافه مانند:
"نام دانشجو را وارد کنید" یا "اطلاعات درس را وارد کنید"، خودداری نمایید.

نمونه ورودی‌ها:

```
3
Physics 3111 3 2
Math 3112 2 5
AP 3113 3 3
2
Dara 9831001 14 2 3111 3112
Sara 9431002 16 3 3111 3113 3112
1
Dara 2 3111 3113
```

نمونه خروجی‌ها:

```
Physics: Sara
Math: Dara Sara
AP: Sara Dara
Dara: Math AP
Sara: Physics AP Math
```

سوال پنجم

شما به عنوان برنامه‌نویسی جاوا در سازمان اجرای مقررات رسانه‌های دیجیتال کشور استخدام شده‌اید. رئیس شما قصد دارد تا تمامی مراکز پخش محتوای دیجیتال کشور را شناسایی کرده و لیست محصولات به

فروش رسیده در آن‌ها را در یک برنامه‌ی جاوا ذخیره کند تا بتواند به راحتی با تشخیص محتوای غیرمجاز، حکم لغو فعالیت این مراکز را صادر کند.

همان‌طور که می‌دانید، هر کشور دارای تعدادی استان است و هر استان شامل تعدادی مرکز پخش؛ مراکز پخش محتوا به مراکز فروش فیلم و سریال، مراکز فروش موسیقی و مراکز فروش بازی‌های رایانه‌ای تقسیم می‌شوند.

هر مرکز پخش محتوا شامل نام، تعدادی محتوا که اسم و قیمت‌های مختلفی دارند و مجوز فعالیت است. در صورت عدم صدور مجوز فعالیت یا صدور حکم لغو فعالیت، مرکز پخش محتوا قادر به فروش محصولی نیست. همچنین، خود سازمان مقررات نیز دارای فهرستی از نام محصولات غیرمجاز است.

برای نوشتن این برنامه به طوری که رئیس حقوق شما را پرداخت کند، باید با استفاده از طراحی شی‌گرا و **Collection های جاوا** ساختارهای ذکر شده در بالا را پیاده کرده و فرضیات زیر را رعایت کنید:

- کشور حداقل ۱۰ استان دارد (لازم نیست اسم همه‌ی استان‌ها را بنویسید، صرف داشتن اسم‌های منحصر به فرد کافی‌ست)
- هر استان حداقل ۳ مرکز پخش محتوا دارد. (حداقل از هر نوع مرکز یکی)
- هر مرکز پخش تعداد دلخواهی محتوا دارد که قیمت آن‌ها حتماً عدد صحیحی مثبت و بالاتر از ۵۰۰ تومان است.

۱. ابتدا در ذهن خود type های مناسب برای موجودیت‌های سیستم را بیابید و سپس کلاسی برای کشور بنویسید که وضعیت شرح داده شده را مدل‌سازی کند. کلاس کشور طوری طراحی شود که در خود Collection ی برای نگهداری اطلاعات مورد نیاز داشته‌باشد.

متدهای مورد نیاز برای این کلاس به صورت زیر هستند:

- متدی برای اضافه کردن یک استان به کالکشن موجود در کلاس کشور.
 - متدی برای اضافه کردن یک مرکز پخش جدید به یکی از استان‌های موجود در کشور.
 - متدی برای اضافه کردن یک محصول به یکی از مراکز پخش یکی از استان‌های موجود در کشور.
 - متدی برای تشخیص محصول غیرمجاز قبل از اضافه کردن آن محصول به مرکز فروش.
- (راهنمایی: Collection اصلی کلاس کشور شما احتمالاً ۳ لایه‌ی تودرتو خواهد داشت!)

۲. در تابع main، شبیه‌سازی‌ای برای اضافه شدن استان‌ها، مراکز فروش و محصولات به یک

instance از کشور بنویسید. این تابع باید در کلاس CultureMinistry قرار بگیرد.

در شبیه سازی، ابتدا استان‌ها و مراکز پخش آن‌ها به صورت object هایی ایجاد شده و سپس مراکز پخش محصولاتی را اضافه می‌کنند. در صورت اضافه شدن محصولی غیرمجاز، مرکز در همان لحظه پلمپ می‌شود.

در انتها نیز لیستی از استان‌ها به همراه نام مراکز فروش آن‌ها و مجموع قیمت محصولات هر مرکز فروش نمایش داده شود. (راهنمایی: از متد toString استفاده کنید).

نکته: برای این سوال، شما تنها مجاز به نوشتن کلاس برای کشور و انواع مراکز فروش هستید و بقیه‌ی موجودیت‌های سیستم یعنی استان‌ها و محصولات باید با Collection‌ها پیاده‌سازی شوند. (راهنمایی: به مستندات کلاس Map و کلاس‌های مربوط به آن برای ساخت استان‌ها و محصولات مراجعه کنید.)

(راهنمایی: در زیر یک نمونه از استفاده‌ی کالکشن‌ها به صورت تودرتو آورده شده که جدول ضرب را در یک ArrayList از ArrayList‌ها قرار می‌دهد.)

(برای پیاده‌سازی این تمرین نیازی به مفاهیم ارث‌بری نیست. تاکید ما بر استفاده از کالکشن‌ها است)

```
import java.util.ArrayList;

class Main {
    Run | Debug
    public static void main(String[] args) {
        ArrayList<ArrayList> matrix = new ArrayList<ArrayList>();
        for (int i = 1; i <= 10; i++)
        {
            ArrayList<Integer> row = new ArrayList<Integer>();
            for (int j = 1; j <= 10; j++) {
                row.add(i * j);
            }
            matrix.add(row);
        }
        System.out.println(matrix);
    }
}
```