

# مکتب شریف

اولین بوتکمپ آموزشی - استخدای ایران



تمرین دهم

مکتب شریف





## گیت:

برای تمرین های جاوا خود از امکانات git مانند branch, pull, push, commit استفاده کنید.

تمرین های خود را در remote repository قرار دهید و آدرس آن را همراه فایل تمرین ارسال کنید.

نکته: قبل از شروع کد نویسی جاوا حتما git init بکنید و مرحله به مرحله کدهایتان را commit کنید. کامیت های معنا دار بزنید. (میتوانید از یک برنچ استفاده کنید)

-----

1. الف) با استفاده از HashMap برنامه ای بنویسید که برای هر کلمه ی ورودی از سمت کاربر، کلیه کلماتی که می شود فقط با حروف آن کلمه ساخت را در خروجی چاپ کند. برای مثال برای کلمه CAT می توان ACT, TCA, TAC, CTA را ایجاد کرد. هر حرف باید دقیقا به همان تعدادی که در کلمه اصلی وجود دارد تکرار شود.

ب) اگر کاربر دو کلمه وارد کرد، مثلا اگر کاربر کلمه CAT و کلمه ACT را وارد کرد، اگر دقیقا شرایط گفته شده را داشته باشند، پیغام Pass و در غیر اینصورت Fail را در خروجی چاپ کند.



2. الف) با ایجاد دو TreeSet آنها را با 10 کارکتر تصادفی از حروف 'a' تا 'z' پر کرده و محتوای این دو مجموعه را در خروجی نمایش دهید.
- ب) متدی بنویسید که دو مجموعه فوق، ورودی آن باشد و اجتماع آنها را پیدا کند و در خروجی نتیجه را چاپ کنید.
- ج) با استفاده از یک متد دیگر اشتراک آنها را پیدا کرده و در خروجی نتیجه را چاپ کنید.

3. برنامه ای بنویسید که در آن یک متد، پارامتری به عنوان ArrayList از اعداد صحیح بپذیرد و زوج مرتب های "نامناسب" آن لیست را حذف کند.
- زوج مرتب نامناسب دو عنصر در همسایگی یکدیگر هستند که عنصر سمت چپ زوج مرتب بزرگتر از عنصر سمت راست باشد. هر عضو سمت چپ زوج مرتب، دارای اندیس زوج در لیست می باشد و هر عضو سمت راست زوج مرتب دارای اندیس فرد در لیست است.
- برای مثال:

{1, 3, 7, 4, 3, 6, 5, 8, 5, 5, 2, 9, 7, 3}

که ما می توانیم آن را به صورت زوج مرتبهای زیر در نظر بگیریم:

{1, 3}, {7, 4}, {3, 6}, {5, 8}, {5, 5}, {2, 9}, {7, 3}

از آنجایی که در زوج های {1, 3}, {3, 6}, {5, 8}, {2, 9} عنصر اول از عنصر دوم بزرگتر است، پس این زوج ها "نامناسب" نیستند و کاندید حذف شدن

هستند، فلذا پس از فراخوانی متد مذکور، لیست به صورت زیر باید تغییر کند:

{7, 4, 5, 5, 7, 3}

اگر لیست دارای طول فرد می باشد، آخرین عنصر را به عنوان یک عضو از زوج مرتب در نظر نمی گیریم و خودش هم به عنوان یک عنصر "مناسب" نیست. بنابراین باید توسط متد شما حذف شود.

اگر لیست خالی به متد ارسال شود، این لیست باید در پایان فراخوانی متد گفته شده، همچنان خالی باشد.

**توجه: مجاز به استفاده از آرایه ها، سایر لیست ها یا ساختمان های داده برای حل این مسئله نمی باشید و فقط باید از ArrayList استفاده کنید.**

**4. کلاس HashMap** را به گونه ای پیاده سازی کنید که شامل ویژگی های زیر باشد (حق استفاده از خود کلاس را ندارید و باید کامل پیاده سازی شود):

- قابلیت Put کردن به HashMap را داشته باشد.
- قابلیت بررسی شامل کلید خاصی بودن را داشته باشد.
- قابلیت بررسی خالی بودن HashMap را داشته باشد.
- قابلیت بدست آوردن همه ی مقادیر را داشته باشد.
- قابلیت جایگزین کردن در HashMap را داشته باشد.

**5. برنامه های بالا را در قالب یک پروژه زده (در قالب کلاس های متفاوت) و پس از اتمام کارتان آن را تبدیل به یک کتابخانه کرده و به عنوان maven dependency در پروژه دیگری آنها را تست کنید.**



## نکات

- نام **فایل** ارسالی خود را به این صورت قرار دهید: Name\_hw9\_maktab96
- به عنوان مثال Mohammad\_Ali\_Kargar\_hw10\_maktab96.
- در صورتی که تمرین شامل چند فایل و فولدر می باشد حتماً آنها را قالب یک فایل فشرده شده تجميع کنید.
- در صورت لزوم یک فایل word به عنوان توضیح در کنار کدهای خود قرار دهید.
- در صورتی که سوالی دارید **در کارتابل گروهی** خود از مربیان بپرسید.
- پس از اتمام تمرین آن را در ریپوزیتوری ای که مربیان مکتب به آن دسترسی دارند قرار دهید.