به نام خدا

پروژه درس برنامه‌نویسی پیشرفته (فلاتر)

عنوان پروژه: طراحی و پیاده‌سازی اپلیکیشن دانشجویار (اپلیکیشن جامع مدیریت دانشجویی)

شرح کلی پروژه:

پروژه این ترم درس AP پیاده‌سازی اپلیکیشن مدیریت جامع دانشگاهی است، که شامل دو بخش مدیریت دانشجو و مدیریت استاد می‌باشد.
قصد داریم برنامه‌ای بسازیم که هر دانشجو امکان مشاهده وضعیت درسی و تحصیلی خود، کلاس‌های ثبت‌نامی، ددلاین‌های پیش رو، آخرین اخبار و ... را داشته باشد و از طرفی هر استاد و یا تی‌ای بتواند وضعیت دانشجویان هر درس خود را مشاهده کند و تمرین یا پروژه‌ای برای آنان منتشر کند و تنظیمات مربوط به کلاس خود را مدیریت کند.
این پروژه به صورت ترکیبی با جاوا و فلاتر می‌باشد و شامل بخش‌های متفاوتی است که به صورت کامل و جامع توضیح می‌دهیم.
پروژه شما از دو بخش cli و اپلیکیشن که با فلاتر پیاده‌سازی می‌شود تشکیل شده است. بخش cli و بک‌اند پروژه (که بخشی از آن را در مینی پروژه پیاده‌سازی کردید) با زبان جاواست.

ابتدا بریم سراغ **اپلیکیشن**:

اپلیکیشن **دانشجویار** از هفت صفحه‌ی اصلی تشکیل شده است. که صفحه به صفحه توضیح داده می‌شود.

صفحه LogIn/ Sign Up:

در این صفحه کاربر با وارد کردن نام کاربری یا شماره دانشجویی و رمز عبور وارد حساب خود می‌شود و می‌تواند وارد صفحات اصلی برنامه بشود و از امکانات اپلیکیشن استفاده کند.

در صورتی که کاربر عضو جدید باشد و حساب کاربری نداشته باشد، باید امکان ثبت نام برایش فراهم گردد.

توجه شود که در صورتی که کاربر لاگین کند، user کاربر باید با بک‌اند ارتباط برقرار کند و بررسی کند کاربر وجود دارد یا خیر. در صورت وجود داشتن کاربر باید وارد صفحه اصلی (سرا) شده و اطلاعات کاربر حفظ شده باشد.

در صورتی که کاربر رمز اکانت اشتباهی وارد کند، باید خطای مناسبی نمایش داده شود. (به طور مثال استفاده از toast notification برای نمایش خطا)

- توجه: طراحی این صفحه کاملا به عهده‌ی خودتان هست و هیچ طرح نمونه‌ای در فیگما ندارد!

در صفحه Sign up امکان ثبت‌نام باید فراهم گردد و موارد گفته شده برای ایجاد حساب جدید را از کاربر دریافت کند.

نام کاربری: در صورت تکراری بودن، خطای مناسب نمایش داده شود.

رمز عبور: باید حداقل دارای 8 حرف باشد و حداقل شامل یک حرف بزرگ و یک حرف کوچک و یک عدد باشد و نباید دقیقا شامل خود نام کاربری باشد. (از ریجکس استفاده گردد) و همچنین قابلیت visible کردن آن فراهم باشد.

شماره دانشجویی: صرفا از عدد تشکیل شده است.

راهنمایی: برای آشنایی با قالب ریجکس کمی جستجو کنید. پیشنهاد می‌شود بررسی معتبر بودن یا نبودن را در فلاتر و هنگام دریافت هر بخش انجام دهید.

راهنمایی: نام‌های کاربری و رمزهای عبور باید در فایل (با کمک جاوا) ذخیره گردد. پیشنهاد می‌شود ویدئوی مربوطه‌ی ضبط شده (شیوه‌ی ذخیره‌سازی داده‌ها در پایگاه‌های داده) را مشاهده کنید.

صفحه‌ی اطلاعات کاربری:

در این صفحه اطلاعات ثبت شده در صفحه‌ی ثبت‌نام کاربر جدید و اطلاعات پیش‌فرض نمایش داده می‌شود.

امکان تغییر هر یک از این داده‌های ثبت شده برای کاربر فراهم می‌باشد.

در این صفحه امکان حذف حساب کاربری قرار داده شده است. با انتخاب این گزینه حساب کاربری شخص حذف می‌شود و وارد صفحه‌ی ثبت‌نام می‌شود.

- نکته: در صورت حذف حساب کاربری، تمام اطلاعات مربوط به کاربر از فایل‌ها و پایگاه داده حذف شود.

صفحه‌ی سرا:

همانگونه که از نام "سرا" پیداست، این صفحه همان home page ماست.

کاربر در این صفحه فعالیت خاص و جدیدی انجام نمی‌دهد و صرفا خلاصه‌ای از سایر فعالیت‌ها و سنجه‌ها را مشاهده می‌کند.

در بخش اول خلاصه‌ای از تعداد تمرین‌ها، امتحان‌ها و ددلاین‌های انجام نشده و بهترین و بدترین نمرات را نمایش می‌دهد.

امتیازی: مدت زمان باقی‌مانده تا اتمام همه‌ی تمارین و کارها را نمایش دهد.

راهنمایی: در مینی‌پروژه ساختار نگاشتی (map) برای ذخیره‌سازی و ساختاردهی نمرات هر دانشجو پیاده‌سازی کردید. با مرتب‌سازی نمرات ذخیره در نگاشت، بیشترین و کمترین نمره را مشخص کنید.

در بخش بعدی تعدادی از کارهای پیش‌روی امروز نمایش داده می‌شود.امکان حذف و انجام هر کار فراهم آید.

در بخش بعدی گزارش کلی از تمارین انجام شده‌ای که هنوز ددلاین آن فرا نرسیده است داده شود.

آیکونی را در صفحه‌ی سرا پیاده‌سازی کنید که با انتخاب آن وارد صفحه‌ی اطلاعات کاربری شویم. (پیاده‌سازی و طراحی آن به هر شکل امکان‌پذیر است.)

صفحه‌ی کارا:

این صفحه مانند todo list ساده‌ای می‌باشد، کارهای پیش‌رو امروز را نمایش می‌دهد.

هر کار را می‌توان حذف کرد و یا انجام داد. کارهای انجام شده از کارهای پیش‌رو مجزا ست و طراحی‌تان به گونه‌ای باشد که تمایز این دو مشخص باشد. (برای درک بهتر به طرح فیگما مراجعه گردد.)

در این صفحه یک FAB (با Floating Action Button در اسلاید ویجت‌ها به عنوان فرزند scaffold آشنا شدیم!) برای افزودن کار جدید پیاده‌سازی کنید.

با انتخاب آن pop upی باز می‌شود و عنوان کار جدید و زمان آن را (هم تاریخ و هم ساعت) مشخص کنید.

راهنمایی: برای این کار از Date - Time در فلاتر کمک بگیرید.

امتیازی: برای هر کار یادآور تنظیم کنید. (به طور مثال دو ساعت پیش از فرا رسیدن زمان آن کار localnotificationی را به دستگاه ما ارسال کند.)

راهنمایی: برای پیاده‌سازی این بخش می‌توانید از لایبرری و پکیج flutter_local_notification استفاده کنید. (**پیوست شماره‌ی 1**)

صفحه‌ی کلاسا:

در این صفحه همه‌ی کلاس‌های ترم جاری در قالب کارت‌هایی نمایش داده می‌شود. (به‌عنوان مثال استفاده از card widget)

هر کارت شامل اطلاعات کلی راجع‌به هر درس و نام استاد هر درس می‌باشد. (عمده‌ی اطلاعات مورد نیاز در مینی پروژه پیاده‌سازی شده است.)

امکان افزودن کلاس جدید فراهم شود. هر کلاس در هر ترم شماره‌ی درس منحصر به فردی دارد که با وارد کردن آن، کاربر دانشجو به آن کلاس اضافه می‌شود.

در صورت عدم وجود کلاس گفته شده، خطاب مناسب نمایش داده شود.

- توجه شود که نام آن شخص باید به اسامی دانشجویان آن کلاس اضافه شود. لذا هر دانشجو امکان دریافت نمره دارد و ممکن است دانشجوی برتر آن درس بشود. (نحوه‌ی محاسبه و پیاده‌سازی نمرات به عهده‌ی خودتان است.)
- مشخصات هر درس، مانند نام و عنوان درس، نام استاد، تعداد ددلاین‌های فعال و ... مربوط به بخش cli و تحت ترمینال پروژه است و مربوط به اپلیکیشن نمی‌شود. ادمین یا هر استادی که درس جدیدی ایجاد کند، شماره‌ی درس متمایزی باید داشته باشد.

امتیازی: روز و ساعت هر کلاس، هنگام ایجاد هر کلاس جدید در cli (که جلوتر در مورد آن توضیح می‌دهیم) از کاربر ادمین یا استاد دریافت شود.

امتیازی: بر اساس زمان‌های هر کلاس، برنامه‌ی هفتگی ویژه‌ی هر دانشجو نمایش داده شود.

راهنمایی: پیاده‌سازی برنامه‌ی هفتگی سخت و پیچیده نیست و صرفا با ذخیره داشتن Date time های نمونه برای هر کلاس و استفاده از پکیج‌ها و لایبرری‌هایی مانند time_planner به سادگی درست می‌شود.
(پیوست شماره 2)

صفحه خبرا:

این صفحه شامل چندین تب مختلف و متفاوت است.
هر خبر به صورت یک کارت نمایش داده می‌شود و پیاده‌سازی صفحات دیگر کاملا به عهده و اختیار خودتان است. (توصیه می‌شود جهت تکمیل بودن برنامه‌ی دانشجویاری که پیاده‌سازی کردید تمامی تب‌ها پیاده‌سازی شود و حاوی اطلاعات مربوطه باشد.)

امتیازی: با انتخاب هر خبر از اخبار یا رویداد، صفحه‌ی وبی به دلخواه خودتان به صورت web view نمایش داده شود. مانند صفحه اخبار سایت دانشگاه/دانشکده. (راهنمایی: استفاده از پکیج webview_flutter)
(پیوست شماره 3)

امتیازی: در صورت تمدید هر تمرین (که توسط استاد یا ادمین در cli صورت می‌گیرد) یادآوری محلی (local notification) به کاربر دانشجو ارسال شود. در صورت پیاده‌سازی notification در بخش قبلی پیاده‌سازی این بخش بسیار ساده است.

امتیازی و راهنمایی: برای پیاده‌سازی بخش‌های دیگر حتما به تاریخ امروز نیاز داریم. در صفحه تولدهای امروز نام دانشجویانی که تولد آن‌ها امروز است نمایش داده می‌شود. برای این قسمت فایل جداگانه‌ای در نظر بگیرید که نام هر دانشجو و تولد او در آن ذخیره شده باشد. در بک‌اند با مقایسه‌ی تاریخ امروز با تاریخ تولد هر فرد، تولدهای امروز پیدا می‌شود و برای اپلیکیشن فرستاده می‌شود. برای راهنمایی می‌توانید تولدها را به صورت نگاشتی که کلید آن رشته‌ای از نام هر فرد و مقدار آن هم از جنس Date type و تاریخ باشد. برای روشن‌تر شدن پیاده‌سازی این بخش کمی جستجو کنید و کدها و مثال‌های آماده‌ی موجود در اینترنت را بررسی کنید.

صفحه‌ی تمرینا:

در این صفحه تمرین‌هایی که برای هر درس ثبت شده و مهلت تحویل آن‌ها در روز مشخص شده است (پیش فرض امروز) نمایش داده می‌شود. امکان تغییر روز باید پیاده‌سازی شود. (توصیه می‌شود از Date picker استفاده کنید.)
تمرین‌هایی که انجام شده‌اند یا زمان تحویل (ددلاین) آن گذشته از سایرین باید تفکیک گردد.
با انتخاب هر تمرین pop-upی برای نمایش جزئیات آن تمرین نمایش داده شود. (جزئیات هر تمرین و داده‌های مورد نیاز را در مینی‌پروژه پیاده‌سازی کردید.)

در جزئیات تمرین عنوان تمرین، روز باقی‌مانده تا ددلاین، مدت زمان تخمینی، توضیحات، نمره و توضیحات تحویل نمایش داده می‌شود. (همانند طراحی فیگما)

- همانند طرح فیگما، تنها مدت زمان تخمینی و توضیحات و توضیحات تحویل توسط کاربر (دانشجو) قابل تغییر است.
- ویژگی (فیلدی) به نام estimated_time به هر تمرین اضافه کنید و هر استاد هنگام تعریف هر تمرین جدید مقدار پیش‌فرض اولیه‌ای به عنوان سازنده (constructor) به آن می‌دهد.که این فیلد بعدا توسط دانشجو، بر اساس نظر خودش و یا وقت صرف شده، می‌تواند مقدار آن را تغییر دهد.

امتیازی: امکان بارگذاری فایل pdf به عنوان تحویل تمرین فراهم شود. نیازی به ارسال آن به بک‌اند و ذخیره کردن آن در پایگاه داده(فایل) نیست.

امتیازی و پیشنهادی: تمارین پیشرو (تحویل داده نشده) را بر اساس زمان تعریف، نزدیکی ددلاین پیش‌رو و مدت زمان کار کمتر (بر اساس زمان تخمینی باقی‌مانده - estimated time) مرتب‌سازی و نمایش دهید. همچنین در مورد شیوه مرتب‌سازی و پیاده‌سازی توابع آن گزارش کوتاهی تهیه کنید.
راهنمایی: توصیه می‌شود سه متد برای مرتب سازی بر اساس زمان تعریف (آشنایی با FIFO)، نزدیکی ددلاین پیش‌رو (آشنایی با SJF) و مدت زمان کار کمتر (آشنایی با SRT) در جاوا پیاده‌سازی کنید. برای پیاده‌سازی آسان‌تر می‌توانید از امکانات جاوا 8 که در درس با آن آشنا شدید، استفاده کنید. (**پیوست 4**)

پیاده‌سازی cli:

اکثر پیاده‌سازی‌های مربوط به این بخش را در مینی‌پروژه انجام داده‌اید. در این بخش قرار است پیاده‌سازی‌های مینی پروژه را کمی کامل‌تر و ساختار یافته‌تر کنیم. اولا که تمامی توابعی که پیاده‌سازی کردید را مرور کنید و تمامی فیلدها یا متدهای جدیدی که مربوط به کلاس‌های پیاده‌سازی شده‌تان هست را اضافه کنید.
اگر کدی که در مینی‌پروژه تحویل دادید امکان دریافت خطای null pointer دارد آن را درست کنید.
به دلخواه خودتان محیطی در ترمینال ایجاد کنید که در ابتدای اجرا، نقش خود را به عنوان ادمین و یا استاد مشخص کند.
ادمین بالاترین سطح دسترسی را دارد و می‌تواند استاد جدیدی را تعریف کنید. می‌تواند هر درسی را کلا حذف کند. دانشجویانی را از/به کلاس‌ها حذف و اضافه کند. برای هر درسی تمرین تعریف کند و .... پس به صورت خلاصه ادمین بدون هیچ شرطی امکان دسترسی به تمامی متدهای پیاده‌سازی شده را دارد.
هر استاد نیز می‌توان با شماره استادی خود، صرفا به تمامی متدها و ویژگی‌هایی که مربوط به خود و درس‌هایش است دسترسی داشته باشد.
علاوه بر موارد ذکر شده، بعد از هر دستور جدید، بسته به پیاده‌سازی خودتان، اطلاعات جدید را در پایگاه داده‌ی خود (بر اساس فایل متنی) اصلاح کنید و وارد کنید.

پیشنهادی: پیشنهاد می‌شود جهت زیباتر شدن منوی پیاده‌سازی شده، اطلاعات قبلی چاپ شده پاک شود. **(پیوست 5)**

امتیازی: پیاده‌سازی گرافیکی محیط cli با جاوا.

نکات مهم:

- در این پروژه صرفا و حتما باید از دو زبان جاوا و دارت (فریم‌ورک فلاتر) استفاده شود و هیچ زبان یا فریم‌ورک دیگری مجاز نیست!
- جهت دیدن نمونه و آسان‌تر شدن کار شما در پیاده‌سازی نمونه‌ای در فیگما طراحی شده است، که می‌توانید از آن کمک بگیرید.
  اکثر صفحات اصلی در فیگما طراحی شده است.
  طرح‌های فیگما تا حد امکان بسیار ساده (البته کامل) و بدون پیچیدگی عجیبی برای پیاده‌سازی طراحی شده است.
  در فیگما امکان خروجی گرفتن از آیکون‌ها، بررسی کد رنگ‌ها، بررسی سایز و نسبت هر کامپوننت به نسبت سایر اجزا مشخص است و پیاده‌سازی شما را بسیار آسان‌تر می‌کند. (در ویدئوی 4 با کار با فیگما آشنا شدید.)
  اندازه‌ی پیش‌فرض هر صفحه در طراحی 360 * 800 در نظر گرفته شده است که بر اساس سایز صفحه‌ی گوشی (چه مجازی، چه فیزیکی) که قرار است روی آن اجرا بگیرید، طراحی را با حفظ نسبت‌ها resize کنید.
- در این پروژه باید از مفاهیمی که در درس برنامه نویسی پیشرفته (سوکت، نتورک، ترد، فایل و مباحث مرتبط با شی‌گرایی) که آموخته اید استفاده کنید. ارتباط میان سرور و اپلیکیشن تماما باید با Socket باشد.

  برای فاز امتیازی استفاده از api، فایربیس و ... مجاز است اما باید هر دو ارتباط موجود باشد. (به طور مثال در گیت‌هاب پروژه‌ی خود دو برنچ نهایی داشته باشید، یکی با api و دیگری که نسخه‌ی پایه‌ی آن است با سوکت.)

  ذخیره‌سازی در فایل نیز اجباری است در صورت پیاده‌سازی ذخیره‌سازی داده‌ها در پایگاه‌داده‌ای غیر از فایل باید هر دو شیوه موجود باشد. (به طور مثال در برنچ دیگری در گیت‌هاب)
- سرور برنامه باید توانایی پاسخگویی به چندین کاربر مختلف در زمان واحد را داشته باشد. (پیاده‌سازی با ترد)
- با بسته شدن برنامه سرور، هیچ داده ای نباید پاک شود و با اجرای دوباره سرور داده‌ها باید دوباره بارگذاری شده و از حالت قبلی ادامه یابد. (خواندن داده‌های مربوط به هر کاربر از پایگاه داده)

- توجه داشته باشید که اجباری برای پیاده‌سازی صددرصدی اپ‌های ذکر شده نیست و صرفا پروژه شما باید دارای امکانات و صفحات خواسته شده باشد و هرگونه **خلاقیت اضافه** (چه در قابلیت‌های نرم‌افزار و چه در طراحی رابط کاربری) مجاز می‌باشد و در صورت صلاحدید تحویل گیرنده، به عنوان امتیاز مثبت منظور می گردد.
- همانگونه که می‌دانید پروژه در قالب گروه‌های دو نفره انجام می‌شود. پیش از شروع پیاده‌سازی، بین هر دو عضو گروه تقسیم کاری صورت گیرد و مشخص باشد که هر فرد چه بخش‌هایی را قرار است پیاده‌سازی کند. (برای تقسیم‌بندی و تسک‌بندی پروژه از ابزارهای زیادی می‌توانید استفاده کنید. از خود گیت‌هاب برای تسک‌بندی میتوانید کمک بگیرد. یکی از سایت‌هایی که کارایی بالایی دارد و استفاده از آن آسان است، Trello نام دارد.)

  در هنگام تحویل پروژه تسلط هر فرد به کلیات بخش‌هایی که هم‌گروهی او پیاده‌سازی کرده سنجیده می‌شود و تسک‌بندی و تقسیم بندی پروژه نیز توسط تحویل گیرنده مورد بررسی و ارزیابی قرار می‌گیرد.

- در کارگاه‌های آموزشی برگزار شده با git و Github آشنا شدید. استفاده از GitHub اجباری است و میزان استفاده از گیت (استفاده یا عدم استفاده، تعداد کامیت و کامیت‌های هر فرد و برنچ‌ها) شامل نمره اختصاصی است و عدم استفاده به منزله دریافت نکردن نمره آن بخش است. (پیشنهاد می‌شود در هر مرحله برای پیشرفت کار، تمامی تغییرات خود را با کامیت‌های مناسب در ریپوی خود پوش کنید.)

  با نحوه‌ی درست کامیت زدن آشنا شده‌اید. (**پیوست شماره 6**)

- ریپوی شما تا پایان تحویل پروژه باید به صورت Private باشد و پروژه را با یکی از اعضای گروه حل تمرین پروژه که مشخص می‌گردد تا پایان تحویل پروژه در گیتهاب جهت بررسی روند پیشرفت کارتان یا احیانا جهت کمک گرفتن یا رفع باگ‌های احتمالی collaborate کنید.
- اضافه کردن توضیحات و تصاویر مناسب و گزارشات تهیه شده (امتیازی) از اپلیکیشن به repository پروژه در گیتهاب الزامی است.
- تمیزی کد شما (شامل نام‌گذاری‌ها و کامنت‌گذاری‌ها و ...) مورد ارزیابی قرار می‌گیرد.
- بخش قابل توجهی از نمره‌ی دریافتی از پروژه مربوط به تسلط شما بر پروژه است. در صورت مشاهده‌ی هرگونه تقلب نمره‌ی بخش پروژه‌ی شما صفر می‌شود.

فازبندی پروژه

1. فاز اول پروژه: (6 خرداد)
   - تکمیل پیاده‌سازی‌های مربوط به مینی‌پروژه
   - پیاده‌سازی cli و منوی کنسول جاوا (در صورت طراحی گرافیکی، یعنی خارج از محیط ترمینال، ددلاین پیاده‌سازی منو به انتهای فاز دوم پروژه موکول می‌شود.
   - پیاده‌سازی و طراحی صفحه‌ی login - signup - اطلاعات کاربری در فلاتر
   - پیاده‌سازی توابع مربوطه با آن (در بک‌اند) و ذخیره‌سازی در پایگاه داده (فایل) در بک‌اند پروژه (جاوا)

   ! توجه: در این فاز نیازی به اتصال و ارتباط بک‌اند و فرانت‌اند نیست.
   - پیاده‌سازی چهار کلاس طراحی شده (student - teacher assignment - course) در فلاتر.

   !توجه: پیاده‌سازی کلاس‌ها، شامل فیلدها و سازنده‌ها، کاملا مطابق با مینی‌پروژه باشد.

2. فاز دوم پروژه: (12 تیرماه)
   - پیاده‌سازی بک‌اند و فرانت‌اند سایر صفحات اصلی و پیاده‌سازی جزئیات
   - اتصال بک‌اند به فرانت‌اند
   - پیاده‌سازی مولتی تریدینگ برای پشتیبانی از استفاده همزمان چند کاربر

3. فاز سوم: (17 تیرماه - تحویل حضوری)
   - زیباسازی نهایی و پیاده‌سازی بخش‌ها و جزئیات امتیازی
   - تمیز کردن کد
   - تکمیل readme گیت‌هاب و تهیه‌ی گزارش (از بخش‌های امتیازی مشخص شده)

4. فاز چهارم (امتیازی و پیشنهادی): (17 تیرماه - تحویل حضوری)
   - مشاهده ویدئوها و یادگیری api یا فایربیس - اتصال فلاتر و جاوا با API
   - تهیه‌ی گزارش و مقایسه دو شیوه‌ی از نظر پیاده‌سازی آسان‌تر و عملکرد بهتر
   - ذخیره‌سازی داده‌ها در پایگاه داده‌ای غیر از فایل
   - تهیه‌ی گزارش و مقایسه دو شیوه‌ی پیاده‌سازی شده از نظر پیاده‌سازی آسان‌تر و عملکرد بهتر و بررسی سود و زیان هر کدام از دو روش

پیوست:

1. برای آشنایی با local notification در فلاتر و نحوه‌ی استفاده از آن لینک‌های زیر مفید است:

- https://fluttergems.dev/packages/flutter_local_notifications/
- https://www.youtube.com/watch?v=iKxrt4ASR5Y
- https://maneesha-erandi.medium.com/add-a-reminder-to-your-app-with-flutter-local-notifications-dfb2e5120499
- https://www.youtube.com/watch?v=5d9dAbm9Ln4
- https://fluttergems.dev/packages/onesignal_flutter/

2. آشنایی با plannerها برای برنامه هفتگی در فلاتر:

- https://fluttergems.dev/packages/syncfusion_flutter_calendar/
- https://fluttergems.dev/packages/time_planner/

3. استفاده از web view page در فلاتر:

- https://fluttergems.dev/packages/webview_flutter/
- https://fluttergems.dev/packages/flutter_inappwebview/
- https://www.youtube.com/watch?v=wPf-7rrng-8
- https://www.youtube.com/watch?v=j5JamZ-uBYY

4. در درس سیستم عامل در مبحث cpu scheduling با انواع الگوریتم‌های زمان‌بندی آشنا می‌شوید. در آن‌جا پراسس‌های سیستم‌عامل بر اساس الگوریتم‌های زمان‌بندی cpu را میان پراسس‌ها تقسیم‌بندی می‌کند.
نام الگوریتم‌های ذکر شده بدین ترتیب است:
    - FIFO : first in first out
    - SRT / SRTF: shortest remaining time first
    - SJF: shortest job first

- ما از شما میخواهیم مشابه این الگوریتم‌ها را بر روی لیست تمارین پیاده‌سازی کنید. توصیه می‌شود پیاده‌سازی را در جاوا و با استفاده از امکانات جاوا 8 انجام دهید.
- اگر ذخیره سازی تمارین در لیست باشد، خودش به صورت پیش‌فرض ترتیبی است و حالت FIFOیی دارد.
- بر اساس Date time (اختلاف زمان کنونی با زمان ددلاین) و مرتب سازی لیست بر اساس کمترین تفاضل بدست آمده، حالت SJF را برای ما می‌سازد.
- در صورت مرتب‌سازی لیست تمارین، در لیست جدیدی بر اساس زمان تخمینی باقی‌مانده حالت SRT را میتوانید پیاده‌سازی کنید.

- در مجموع با فراخوانی این توابع باید سه لیست داشته باشید که احتمالا ترتیب تمارین در هر کدام متفاوت است. این سه لیست به فلاتر فرستاده می‌شود و در قالب حالت‌های سورت‌بندی نمایش داده می‌شود.
- پیشنهاد می‌شود گزارشی درباره‌ی نحوه‌ی پیاده‌سازی و مختصری درباره‌ی 3 شیوه‌ی نامبرده تهیه کنید.

5. جهت پاک کردن کنسول در جاوا می‌توانید از دستور زیر استفاده کنید:

```
System.out.print("\033[H\033[2J");
System.out.flush();
```

جهت چاپ رنگی هم به لینک زیر مراجعه کنید.

- https://www.tutorialspoint.com/how-to-print-colored-text-in-java-console

6. آموزش نحوه‌ی درست کامیت زدن:

- https://www.conventionalcommits.org/en/v1.0.0/

موفق باشید:)

بهار 1403