پروژه درس برنامه نویسی پیشرفته

بخش دوم

زمستان 1399 - دانشکده علوم ریاضی

دانشگاه صنعتی شریف

تیم طراحی

مجتبی استواری

سروش تابش

سید محمد ترابی

نیما خداویسی

حسین رحمانی

محمد مهدی زارع

سید امیر محمد سادات شکوهی

محمدعلی علما

نیما علیزاده

غزل فراهانی

سید عرفان موسویان

هلیا یزدان یار

معرفی پروژه

در این پروژه شما باید یک شبکه اجتماعی طراحی کنید. پیشنهاد می‌شود قبل از شروع به کار فایل توضیحات کلی پروژه که قبلا در اختیار شما قرار گرفته است را مطالعه کرده باشید. همچنین خوب است با برخی از شبکه‌های اجتماعی نظیر توییتر آشنا باشید.

اهداف بخش دوم پروژه

- آشنایی با گرافیک در جاوا
- استفاده از اصول طراحی شی‌گرا
- کار با resource و فایل
- اصول کلین کد

رفع اشکال

برای این بخش از پروژه یک جلسه توجیهی برگزار می‌شود که زمان آن متعاقبا اعلام خواهد شد. قبل از جلسه توجیهی، حتما داک را مطالعه کنید. همچنین شما می‌توانید برای رفع پرسش‌ها، اشکال‌ها و ابهام‌های خود از طریق سایت کوئرا با تیم طراحان پروژه در ارتباط باشید.

ارزیابی و روش تحویل

- این بخش جداگانه از بخش های دیگر ارزیابی و به صورت آنلاین تحویل گرفته می‌شود.
- هر بخش پروژه باید حداکثر تا تاریخی که به شما اعلام شده است در کوئرا آپلود شود. کدی که در زمان تحویل مورد ارزیابی قرار می‌گیرد، کد آپلود شده در کوئرا می باشد. دقت کنید که ددلاین‌های اعلام شده قابل تغییر نمی‌باشند پس برنامه ریزی لازم را برای رساندن پروژه خود به ددلاین‌ها داشته باشید.
- دقت کنید که فقط کارکردن کد مدنظر نیست و از شما انتظار می‌رود که به صورت اصولی، تمیز و پیشرفته کد بزنید!
- در هر بخش پروژه شما **ملزم** به نوشتن یک فایل توضیحی کامل در مورد کد خود هستید. روش نوشتن این فایل توضیحی در قسمت بعدی نوشته شده است.

نوشتن بخش توضیحات

هر بخش (فاز) پروژه شما نیاز به یک فایل توضیحی است که موارد زیر را شامل می شود:

1. منابع استفاده شده برای پیاده سازی کد شامل
    a. منبع تصاویر و ...
    b. منبع کدها و ...
    c. مشورت های انجام شده
    d. کتابخانه های استفاده شده
2. روش کارکرد کد شما به همراه نقاط قوت و ضعف آن
3. ارائه دلیل برای انتخاب هایی که انجام داده‌اید (مثلاً چرا از یک کتابخانه خاص یا طراحی خاص استفاده کرده اید)

این نوشته به ارزیابی سریع تر و راحت تر پروژه شما کمک فراوانی می‌کند. ترجیحا توضیحات پروژه کوتاه، مختصر و مفید باشد.

موارد غیر مجاز

- عدم تسلط کافی بر کد پروژه
- شباهت بیش از حد دو یا چند پروژه
- واگذاری کامل یا بخشی از پروژه به شخصی دیگر

رخ دادن این اتفاق‌ها برای هیچ فردی قابل پذیرش نیست و در صورت بروز هر کدام از این اتفاق‌ها ممکن است هر تصمیمی در رابطه با ارزیابی فرد گرفته شود.

در صورتی که یکی از این اتفاق‌ها رخ داده باشد لازم است که افراد حتما دلیل این مساله را پیش از تشخیص توسط تیم درس اعلام کنند و دلیل این اتفاق را توضیح دهند. در این صورت فقط ارزیابی مربوط به بخش اعلام شده تحت تاثیر قرار خواهد گرفت.

در صورتی مشاهده یکی از این اتفاق‌ها توسط تیم درس و پیش از اعلام فرد رخ دهد این تیم از فرد درخواست خواهد کرد که در این رابطه توضیح دهند و در صورت قابل قبول نبودن توضیح، فرد مورد نظر موفق به گذراندن درس نخواهد شد.

نکات مهم

- ممکن است در آینده اصلاحاتی در این داک صورت بگیرد یا توضیحات بیشتری برای بعضی قسمت‌ها اضافه شود. سعی می‌شود تغییرات ایجاد شده، با متن‌های گذشته تفاوت بصری داشته باشد تا راحت‌تر قابل تشخیص باشند. این اصلاحات، از طریق راه‌های معمول (مثل ایمیل) اطلاع رسانی **نخواهد شد.** بنابراین سعی کنید از داک به صورت آنلاین استفاده کنید یا در صورت استفاده از قالب pdf ، **به طور مرتب** نسبت به دریافت نسخه تازه اقدام کنید.
- در ادامه، توضیحات کلی مواردی که لازم است در برنامه شما وجود داشته باشد داده شده است. در صورتی که درباره جزییات چیزی توضیح داده نشده، از **خلاقیت** خود استفاده کنید و به هر صورتی که دوست دارید آن قسمت را پیاده سازی کنید.

ذخیره سازی اطلاعات

بدیهی است که یکی از مهم ترین قابلیت‌های یک شبکه اجتماعی (و کلا هر برنامه دیگری) ذخیره سازی اطلاعات است. در شبکه اجتماعی شما نیز، باید اطلاعات کاربران به نحو صحیح ذخیره شود و در مواقع لزوم از آنها استفاده شود.

معمولا در پیاده‌سازی دیتا مدل‌های یک پروژه مشکلات زیر رخ می‌دهد:

- چگونه باید مدل‌های ساخته شده با یک زبان خاص را طوری ذخیره کرد که یک برنامه دیگر بتواند از آن استفاده کند؟
- چگونه باید این مدل‌ها را به صورتی ذخیره کرد که هم برای ماشین خوانایی داشته باشد و هم برای انسان عادی؟

این مشکلات وقتی نمود پیدا می‌کنند که برنامه نیاز دارد داده‌های خود را در فایل، دیتابیس و... ذخیره کند که به زبان برنامه‌نویسی وابسته نیستند. توجه کنید که در بخش‌های آینده پروژه، دیتا مدل‌های جدید (چه قبل از کامپایل و چه در حال اجرا) به برنامه شما اضافه می‌شود. پس طراحی شما باید Abstraction کافی را داشته باشد تا این موارد را مدیریت کند و در هنگام اجرا به مشکل برنخورد.

در کنار این‌ها، شاید لازم شود که مقادیر پیش‌فرض دیتا مدل‌ها را بدون اجرای دوباره‌ی سورس کد تغییر دهید. بنابراین، از شما می‌خواهیم که تمامی دیتا مدل‌ها را به صورت جداگانه در فایل‌هایی ذخیره کنید. بعضی از مقادیر باید به صورت پیش‌فرض وجود داشته باشند و قابل تغییر نیستند اما بعضی دیگر توسط کاربر در آینده ممکن است تغییر کنند. این فایل‌ها در هنگام اجرا بارگذاری می‌شوند و اطلاعات مورد نیاز را به برنامه می‌دهند.

در نهایت انتخاب روش ذخیره‌سازی داده‌ها، به عهده خودتان است. اما استفاده از آبجکت استریم‌ها و آبجکت سریالایزرهای جاوا، در کل پروژه و تمرین‌ها ممنوع است.

لاگ کردن (Logging)

یکی از اصول مهمی که در تولید هر برنامه‌ی با کیفیتی رعایت می‌شود، لاگ کردن رخدادهای برنامه است به طوری که بتوان با خواندن لاگ، به روندهای اجرا شده و تغییرات انجام شده پی‌برد. این قابلیت باعث می‌شود نگهداری و توسعه کد با سرعت بسیار بیشتری انجام شود و همچنین اشخاص دیگری که کد شما را می‌خوانند، زمان کمتری را صرف درک کد کنند که دو نکته کلیدی در تولید نرم‌افزار است.

در این پروژه از شما انتظار می‌رود این اصل را **در تمامی فازها** رعایت کنید و لاگ‌های ساختارمندی ذخیره کنید. به این منظور پیشنهاد می‌کنیم برای تولید لاگ، از کتاب‌خانه های موجود استفاده کنید. این کتاب‌خانه‌ها، مسئولیت ذخیره لاگ، حذف لاگ‌های قدیمی، تولید لاگ به فرمت مشخص (سطح اهمیت لاگ، زمان، کلاس تولید کننده لاگ و...)، ارتباط با دیتابیس و... را بر عهده دارند.

برای مثال می‌توانید از کتاب‌خانه log4j یا از کلاس های آماده جاوا استفاده کنید. در این صفحه می‌توانید آموزش مربوط به کتابخانه log4j را نیز ببینید.

چگونه لاگ کنیم؟

دقت کنید که هدف ما از لاگ کردن، متوجه شدن روند کلی برنامه و اتفاقات رخ داده است و همچنین نباید لاگ کردن، سربار عملیاتی زیادی داشته باشد. برای مثال به هنگام برخورد کردن به خطای زمان اجرا، ذخیره کردن stack trace کار خوبی نیست. گرچه اطلاعات زیادی در اختیار ما قرار می‌دهد ولی بسیار حجیم است و برای درک آن، نیازمند درک کد هستیم. به علاوه، در صورتی که از اتفاقات پیش از رخداد خطا بی خبر باشیم، ممکن است درک خطا را دشوار کند. بنابراین به دنبال داشتن یک خلاصه از اجرای روند برنامه هستیم.

اصول خوبی از لاگ کردن را می‌توانید در این صفحه ببینید. **نمره دهی به طور کلی بر اساس موارد داخل این صفحه خواهد بود. موارد ۱، ۲، ۴، ۶ و۷ مهم‌ترین مواردی هستند که باید رعایت کنید.**

تعدادی از مواردی که می‌توانید در لاگ رعایت کنید:

- چه کلاسی
- چه تابعی
- چه زمانی
- سطح اهمیت
- روی چه داده‌هایی
- در چه شرایط محیطی
- چه نتیجه‌ای

چه چیز را لاگ کنیم؟

از شما انتظار می‌رود چیزهایی مشابه به موارد زیر را لاگ کنید:

- شروع برنامه
- باز کردن و بستن فایل
- ورود یا ثبت‌نام کاربر
- تغییر اطلاعات کاربری
- تغییر یا افزودن اطلاعات
- خطاهای مربوط به منطق برنامه
- اتصال و قطع شدن از دیتابیس (و خطاهای مربوطه)
- وضعیت ترنزکشن های دیتابیس (باز شدن، کامیت شدن، لغو شدن)
- خطاهای معمول زمان اجرا (برای مثال عدم توانایی ایجاد فایل، اشکال در ارتباط با دیتابیس، اشکال شبکه و...

رابط گرافیکی

در این بخش از پروژه شما باید رابط نوشتاری (CLI) طراحی شده در بخش قبلی را با یک رابط گرافیکی جایگزین کنید. سایر جزئیات طراحی شده در بخش اول می‌بایست همچنان در دسترس باشند.

به نکات زیر در مورد رابط گرافیکی توجه کنید:

- در تمامی صفحات برنامه باید دکمه‌ای برای خروج از برنامه موجود باشد.
- در همه صفحات به‌جز صفحه ورود و صفحه اصلی برنامه باید یک دکمه برای رفتن به صفحه‌اصلی و دکمه‌ای برای بازگشت به صفحه قبلی وجود داشته باشد.
- سایر منطق ارتباط بین صفحات، مشابه رابط نوشتاری فاز قبل است.
- طراحی سایر جزئیات صفحات برنامه به عهده شما است. البته توجه داشته باشید که قابلیت‌های گفته شده در فاز قبل و این فاز می‌بایست پیاده شده باشند و جزئیات لازم در هر بخش به نمایش در بیاید.
- لازم نیست رابط کاربری شما زیبا یا دارای انیمیشن باشد اما باید حداقل های یک رابط کاربری را داشته باشد.
- برای الگو گیری می توانید اپلیکیشن های توییتر یا تلگرام (برای بخش پیام رسانی) استفاده کنید.

قابلیت‌های جدید برنامه

در این فاز علاوه بر قابلیت‌های قبلی چند قابلیت جدید نیز باید به برنامه اضافه کنید.

1. حذف توییت گزارش شده

اگر یک توییت بیش‌از n مرتبه به عنوان هرزنامه گزارش شد دیگر نباید در تایم‌لاین و یا اکسپلورر برای کسی نمایش داده شود.

2. پیام رسانی گروهی

در صفحه مربوط به پیام‌رسانی باید امکانی برای ایجاد گروه جدید وجود داشته باشد. سپس هر کدام از اعضاء گروه می‌توانند افرادی را که دنبال می‌کنند را به گروه اضافه کنند. (می‌توانید در بخش تنظیمات حریم شخصی به سلیقه خود تنظیمات متفاوتی برای این‌قسمت در نظر بگیرید. البته این‌کار اجباری نیست.) هر کدام از اعضای گروه با ورود به قسمت مربوط به گروه می‌توانند در آن پیام بگذارند و پیام‌های دیگران را بخوانند.

3. ویرایش و حذف پیام در قسمت پیام رسانی

کاربر باید بتواند پیام‌هایی را که در قسمت پیام‌رسانی (چه شخصی و چه گروهی) فرستاده است را ویرایش کند و یا کاملا آن‌را حذف کند. البته می‌توانید این بخش را طوری پیاده سازی کنید که حذف پیام به دیگران اطلاع داده شود. (مانند پیام‌رسان WhatsApp). همچنین توجه کنید که پیام‌هایی که فوروارد شده اند قابلیت ویرایش ندارند.

4. عکس پروفایل برای کاربران

در قسمت ویرایش مشخصات برای کاربران قابلیت انتخاب و تغییر عکس پروفایل برای کاربران وجود دارد. عکس پروفایل هر کاربر در کنار توییت‌ها و پیام‌های ارسالی آن قابل نمایش است. همچنین با ورود به بخش مشخصات یک کاربر باید عکس پروفایل آن با اندازه بزرگتر نمایش داده شود.

5. ارسال پیام تصویری

هر کاربر می‌تواند در بخش پیام‌رسانی یک فایل تصویری از مکانی دلخواه از حافظه دستگاه انتخاب کند و آن‌را به همراه متنی دلخواه به عنوان یک پیام ارسال کند. ویرایش اینگونه پیام‌ها تنها برای قسمت متنی آن انجام پذیر است.

6. توییت همراه با عکس

مشابه قسمت قبل کاربر می‌تواند به همراه توییت خود یک تصویر نیز به اشتراک بگذارد. توییت‌های تصویری تنها از دو بخش متن و عکس تشکیل شده اند که همواره بخش متنی بالای عکس قرار گرفته‌است.

کلین کد (clean code)

1. شما در این فاز از پروژه باید تا مقداری که در کلاس درس، درس داده شده اصول کلین کد را رعایت کنید.
2. پروژه شما باید از پکیج بندی مناسب برخوردار باشد.
3. از قرارداد های نام گذاری در جاوا برای اسم کلاس ها متدها و فیلدها و ... پیروی کنید. همچنین برای نامگذاری، از اسم های مناسبی استفاده کنید.
4. وظایف کلاس های مختلف مشخص باشد و تا حد امکان این وظایف مینیمال باشند.
5. تا جای امکان، تلاش کنید که ارتباط منطق برنامه و گرافیک آن جدا باشد، تا در صورتی که نیاز به تغییر در منطق داشته باشید، نیاز به تغییر گرافیک نباشد. برای اینکار میتواند از یک سری واسطه بین منطق و گرافیک بهره ببرید و یک پل ارتباطی بین قسمت های مختلف پروژه ایجاد کنید. به طوری که ارتباط قسمت منطق برنامه و گرافیک از طریق این پل ارتباطی باشد. این پل ارتباطی می تواند به صورت مجموعه ای از کلاس ها و اینترفیس ها باشد.
6. در این فاز از پروژه دیتا مدل های شما باید تا حد امکان مستقل از منطق برنامه و روش سیو و لود باشد. به طور کلی کلاس های دیتا مدل های شما باید ساختاری شبیه POJO را داشته باشد.
7. تابع هایی که در هر کلاس می زنید تا حد امکان کوتاه و کلی (general) باشند و قابلیت باز استفاده (reusability) داشته باشند. به طور خاص تا حد امکان کد تکراری نداشته باشید.
8. از فایل های کانفیگ استفاده کنید و از هارد کد کردن پارامتر های ثابت برنامه بپرهیزید. توضیحات این فایل ها در ادامه آمده است.
    a. در این بخش از پروژه شما باید از ابزارهای مربوط به فایل های کانفیگ استفاده کنید. فایل های کانفیگ فایل هایی هستند که ما در آنها پارامترهای مورد نیاز برای اجرای برنامه رو نگه می داریم و از هارد کد کردن(نوشتن آنها در کد اصلی) این پارامتر ها خودداری می کنیم. به عنوان یک مثال کوچک می توان به سایز فریم اشاره کرد. می توان به جای نوشتن سایز فریم در کلاس فریم آن را در فایل کانفیگ نوشت و در هنگام لود شدن برنامه این مقدار را از فایل های کانفیگ خواند.
    b. اما این کار چه مزیتی دارد؟ فرض کنید برنامه ای ساخته اید و آن را در اختیار دوستانتان قرار داده اید. اما به دلایل مختلف(مثلا یکسان نبودن رزولوشن نمایشگر ها) دوستانتان نمیتوانند به درستی برنامه را اجرا کنند. بنابراین شما مجبورید که برای هر کدام از دوستانتان، یک نسخه با ویژگی های مطلوب (مانند رزولوشن مانیتور شخص) کامپایل کنید و برای او بفرستید.(فرستادن سورس کد اصلی نیز، در بسیاری اوقات به دلایل امنیتی گزینه ی مناسبی نخواهد بود.) در این شرایط در صورتی که از کانفیگ استفاده کنید، کافی است که اعداد موجود در آن فایل را تغییر دهید و نیازی به کامپایل کردن مجدد کد نیست!

    برای استفاده و خواندن از فایل های کانفیگ راه های زیادی وجود دارد. برای مثال می توانید خودتان با استفاده از مفاهیم کار با فایل، اقدام به خواندن و نوشتن فایل های کانفیگ کنید، یا از کلاس ها و لایبرری های آماده استفاده کنید.برای نوشتن و استفاده از فایل های کانفیگ، میتوانید از کلاس های جاوا مثل Properties استفاده کنید و یا حتی از کتابخانه های خارجی مثل cfg4j و یا Commons Configuration استفاده کنید یا اینکه به روش دلخواه خود عمل کنید. نکته مهمی که خوب است رعایت کنید، برای کارهای مختلف فایل های کانفیگ مختلف داشته باشید و ساختاری شبیه به پکیج بندی کدتان را در فایل های کانفیگ رعایت کنید.

    مواردی که میتوان آنها را در فایل کانفیگ قرار داد :

    1. اعداد ثابت داخل برنامه، مانند سایز فریم ها و ...
    2. آدرس ها، مانند آدرس عکس ها،موسیقی ها، فونت ها، URL دیتابیس و یا آدرس فایل های جیسون (آدرس فولدر ها)
    3. گرافیک : مختصات و سایز کامپوننت های هر پنل(در صورتی که از روش مختصاتی استفاده کرده باشید.)
    4. آدرس خود فایل های کانفیگ! یک فایل کانفیگ که خود شامل آدرس سایر فایل های کانفیگ باشد و فقط آدرس این فایل را در کد تان قرار دهید. برای اینکه این آدرس هم قابلیت عوض شدن داشته باشد می توانید از آرگومان های ورودی برنامه (آرگومان های ورودی چیست؟) یا environment variable سیستم عامل استفاده کنید.
    5. تمام متن های ثابت برنامه (مانند متن لیبل ها، دکمه ها و ...) را درون فایل های کانفیگ قرار دهید.

با آرزوی موفقیت

تیم درس برنامه نویسی پیشرفته