



دانشکده مهندسی

دانشکدگان فارابی

دانشگاه تهران

پروژه پایانی

درس برنامه سازی پیشرفته

استاد راضیه قیاسی

اردیبهشت 1402

## پروژه سیستم بانکی:

در این بخش از درس قرار است به عنوان پروژه، نرم افزار یک بانک را پیاده سازی کنید. در تمرین های قبلی به عنوان یک طرح اولیه، نمونه ای از این نرم افزار را پیاده سازی کرده اید و اینجا لازم است آن را با جزئیات کامل و عملکرد صحیح تحویل دهید.

در این پروژه، مهم ترین نکته، توانایی شما در طراحی معماری این سیستم است و پس از آن، کیفیت پیاده سازی پروژه و پیروی از اصول مهندسی نرم افزار؛ بنابراین لازم است یک پیاده سازی کامل به همراه یک UML دقیق ارائه دهید تا نمره این بخش را به طور کامل کسب کنید.

در صورتی که با نمودارهای UML آشنایی ندارید، درباره آنها در منابع جستجو کرده و اطلاعات لازم را بدست آورید. نمودار شما باید شامل همه کلاس های موجود در سیستم، اتریبیوت ها و روابط میان آنها باشد.

امکاناتی که برای این پروژه در نظر می گیرید باید به امکانات یک سیستم کامل نزدیک باشد، یعنی امکان ثبت نام به عنوان کاربر (مشتری)، ورود به پنل کاربری، انتقال پول و ... و همچنین امکانات مدیریتی مانند پنل مدیرکل، پنل کارمندان و ... . هر چه پیاده سازی شما به یک بانک واقعی نزدیکتر باشد، نمره بالاتری نیز از این پروژه دریافت خواهید کرد.

حداقل امکاناتی که باید در این پروژه پیاده‌سازی کنید:

1) مدیر بانک: فردی که به تمامی اطلاعات مشتریان، کارمندان و تراکنش‌های انجام شده، دسترسی دارد و می‌تواند اعمال زیر را انجام دهد:

a. مشاهده، افزودن، تغییر و حذف کارمندان (مانند افزایش یا کاهش حقوق)

b. مشاهده، افزودن، تغییر و حذف اطلاعات مشتریان

c. مشاهده، افزودن، تغییر و حذف اطلاعات حساب‌ها

d. مشاهده و تغییر اطلاعات وام‌ها و تراکنش‌ها

e. جستجو و مشاهده لیست کارمندان و مشتریان

f. جستجو و مشاهده لیست تراکنش‌ها و وام‌ها

2) کارمند: فردی است که به اطلاعات حساب‌ها و مشتریان دسترسی دارد و توانایی انجام تراکنش بین حساب‌ها را دارد.

a. جستجو و مشاهده لیست مشتریان و حساب‌ها

b. نمایش لیست حساب‌های هر مشتری

c. انجام تراکنش بین حساب‌ها

d. تایید یا رد وام‌های درخواستی

e. خروجی گرفتن از وضعیت یک حساب، مشتری یا وام

3) مشتری: افرادی که در بانک ثبت‌نام می‌کنند و اعمال زیر را انجام می‌دهند:

a. مشاهده لیست حساب‌هایی که به آنها دسترسی دارد

b. انجام تراکنش و انتقال پول از حسابی که به آن دسترسی دارد به حساب‌های دیگر

c. درخواست حذف حساب موجود یا ایجاد حساب جدید

d. مشاهده تراکنش‌های حسابی که به آن دسترسی دارد

4) حساب: چند نوع حساب مختلف در این بانک وجود دارد.

a. حساب جاری

b. حساب سپرده بلند مدت

c. حساب سپرده کوتاه مدت

5) وام: این بانک می‌تواند به مشتری های خود وام دهد. نرخ سود وام و مدت بازپرداخت آن با توجه به نوع وام متغیر است. به شکل پیشفرض وامی با 20 درصد سود برای این بانک تعریف کنید. سیستم باید ماهانه اقساط وام را از حساب مربوطه برداشت کند.

نکات فنی:

- هر کارمند و مشتری توسط یک شناسه یکتا شناسایی می‌شود (مثل کد ملی) و دارای اطلاعات دیگری همچون نام، نام خانوادگی، تاریخ تولد، جنسیت، تاریخ عضویت و ... می‌باشند. همچنین برای ورود به سیستم از کد ملی و رمزعبور افراد استفاده می‌شود. این اطلاعات باید در فایلی ذخیره شوند و در هنگام احراز هویت، جهت مطابقت دادن با اطلاعات ورودی استفاده شوند.
- در هنگام ذخیره کردن اطلاعات کاربران در فایل، یکی از اشتباهات بزرگ امنیتی که می‌توانید مرتکب شوید ذخیره کردن رمزعبور افراد به شکل خام است (یعنی دقیقاً رمزعبوری که وارد کرده اند را به همان شکل داخل فایل بنویسید). به جای این کار باید هش پسورد کاربران را در فایل ذخیره کنید و هر بار هنگام احراز هویت، هش پسورد ورودی را محاسبه کرده و با هش ذخیره شده مقایسه کنید. در این زمینه می‌توانید با جستجو در منابع، اطلاعات کافی کسب کنید.
- حساب های سپرده یا جاری با توجه به استفاده آنها دارای مشخصات متفاوتی خواند بود، برای مثال، حساب های سپرده تا یک تاریخ خاص اجازه انتقال وجه

یا حذف نخواهند داشت و پس از این مدت باید درصد مشخص شده به آنها اضافه بشود.

- با افزودن، تغییر یا حذف هر یک از موارد بالا، باید تمامی اطلاعات مربوط به این تغییرات دوباره بررسی شود و در صورت نیاز به روزرسانی شود. فرض کنید حسابی را حذف می‌کنید، در نتیجه این حساب می‌بایست از لیست حساب‌های مشتری نیز حذف گردد. دقت کنید که این اطلاعات باید در فایل هم به روزرسانی شوند.
- ترجیحا برای هر یک از موجودیت‌ها یک آیدی عددی در نظر بگیرید که با اضافه شدن هر شی جدید، آیدی جدید برابر با آخرین آیدی اعطا شده بعلاوه یک باشد. لیست کاربران و مشتریان باید بر اساس حروف الفبا مرتب شده باشد و لیست حساب‌ها بر اساس آیدی آن حساب در سامانه.
- جهت جستجو و مرتب سازی نمایش لیست‌ها می‌توانید از توابع آماده‌ی موجود در زبان استفاده کنید.
- برای هر نوع از اطلاعات که نیاز به ذخیره‌سازی دارند، فایلی جداگانه در نظر بگیرید. مثلا اطلاعات مشتریان را در فایل customers، کارمندان را در فایل employees و ... نگهداری کنید. فرمت فایل را با توجه به صلاح دید خود انتخاب کنید. استفاده از فایل‌های مرتب مانند CSV یا JSON نیز نمره بیشتری دارد.
- برای مدیریت اطلاعات و لیست‌ها از هر ساختار داده‌ای که تمایل داشته باشید می‌توانید استفاده کنید (به جز آرایه‌های ایستا). ترجیحا از آرایه پویا استفاده کنید. مستندات فنی و راهنمای استفاده از این ساختمان‌های داده با جستجو قابل یافتن و مطالعه خواهند بود.
- در هنگام اجرای برنامه در اولین قدم لازم است که تمامی فایل‌هایی که برای ذخیره‌سازی استفاده کرده‌اید خوانده شوند و برای هر کدام یک آرایه پویا در

نظر گرفته شود (یا ساختمان داده دلخواه) و اطلاعات موجود در فایل در این آرایه قرار گیرد. تمام عملیات های حذف و اضافه و جستجو و ... باید با استفاده از این آرایه های پویا انجام شوند. اطلاعات تغییر یافته نیز باید دوباره در فایل ها ذخیره شوند؛ این عمل را میتوانید بنا به صلاحدید خود در زمان اتمام اجرای برنامه، در زمان اجرای دستور ذخیره سازی یا در زمان ایجاد هر تغییر اعمال کنید. لازم است موارد مختلفی از جمله تاخیر ایجاد شده با هربار نوشتن در فایل و یا احتمال حذف شدن تغییرات با بسته شدن ناگهانی برنامه را نیز لحاظ کنید و بهترین حالت را انتخاب کنید.

- در قسمت های مختلفی که امکان رخ دادن اکسپشن ها وجود دارد، پیغام های مناسب برای کاربر نمایش دهید. مثلاً اگر سن مشتری جدید از یک عدد مشخص کمتر است، یک خطا به او نشان دهید و اجازه ثبت حساب ندهید.
- کد برنامه را در فایل های مناسب و در پوشه های مرتبط قرار دهید. ساختار پروژه شما شامل نمره خواهد بود. می توانید به ساختار پروژه های بزرگ متن باز موجود در Github و ... مراجعه کنید و همچنین راجع به معماری تمیز و دیزاین پترن ها در اینترنت مطالعه کنید.
- نوشتن مستندات مناسب برای کد، کامنت گذاری های کافی (نه بیش از حد لازم) و ... شامل نمره می باشند. در نظر داشته باشید که مستندات باید به اندازه ای واضح باشند که در صورت سپردن پروژه شما به تیم های دیگر، با مطالعه مستندات فنی شما به اجزای مختلف پروژه و عملکرد هر کدام پی ببرند و امکان توسعه آن را در آینده داشته باشند.
- خط فرمان شما و سبک پیاده سازی آن نیز حائز اهمیت و شامل نمره است. نمایش علائم مختلف در هنگام دریافت ورودی، پرسش از کاربر، نمایش لیست و ... و نیز مرتب و واضح بودن متن های نمایش داده شده در نمره دریافتی شما از این بخش تاثیر گذارند.

- کدنویسی تمیز و رعایت اصول مهندسی نرم افزار در نمره دریافتی شما از این پروژه تاثیر بسزایی دارند. درباره اصول مختلف Clean Code مانند DRY، SOLID، YAGNI و ... ، نام گذاری صحیح متغیرها، کلاس ها و ... مطالعه کنید و سعی کنید پروژه ای که پیاده سازی می کنید از این اصول پیروی کنند. به این منظور میتوانید نگاهی به منابع مرتبط همچون کتاب زیر داشته باشید:

Clean Code: A Handbook of Agile Software Craftsmanship ○

نوشته Robert C. Martin، ترجمه این کتاب به فارسی نیز در Github

موجود می باشد. زبان مثال های کتاب Java می باشد اما به علت شباهت

آن به زبان C++ و نیز یکسان بودن عموم نکات و قواعد، می توانید به

عنوان مرجعی برای یادگیری این اصول از آن استفاده کنید.

#### نکات امتیازی:

- پیاده سازی رابط کاربری گرافیکی (GUI) شامل نمره امتیازی می شود. کامل بودن، زیبایی، راحتی استفاده و تجربه کاربری مناسب نکاتی هستند که در نمره دهی به رابط گرافیکی شما حائز اهمیت هستند.
- استفاده از Git و کار با تیم بر روی بستر Github / Gitlab با توجه به اهمیت بسیاری که در بازار کار و صنعت دارد، شامل نمره امتیازی است. هر چه مدیریت بهتری روی شاخه ها، کامیت ها و ... داشته باشید و اصولی تر از گیت استفاده کرده باشید، نمره بالاتری از این بخش دریافت خواهید کرد.
- نوشتن تست برای برنامه نمره امتیازی به همراه دارد. درباره تست های واحد (Unit Test) و انواع مختلف تست های نرم افزار می توانید در منابع مختلف مطالعه و تحقیق کنید. بر اساس درصد پوشش (Coverage) و اصولی بودن تست های نوشته شده به شما نمره امتیازی داده می شود.
- توسعه به روش TDD شامل نمره امتیازی است. استفاده شما از این روش با توجه به تاریخچه گیت پروژه شما تشخیص داده می شود، بنابراین لازم است

در هر کامیت، تست های واحد پیاده شده به همراه کد های آن واحد قرار داشته باشند تا داوری صحیح صورت بگیرد.

- استفاده از دیگر روش ها و اصول مهندسی نرم افزار طبق صلاحدید داور پروژه می تواند شامل نمره امتیازی شود.

#### نکات مهم ارائه:

- نوشتن پروژه به تنهایی موجب دریافت نمره نمی شود، بلکه لازم است حتما پروژه خود را ارائه دهید. ارائه صحیح و توضیح کامل اصول و تکنیک های استفاده شده شرط اصلی دریافت نمره هر بخش است.
- برای ارائه (در صورت حضوری بودن ارائه ها) حتما باید لپ تاپ به همراه داشته باشید که پروژه را بر روی آن اجرا و ارائه کنید؛ در صورتی که لپ تاپ شخصی ندارید از لپ تاپ دوستان خود استفاده کنید و نیازمندی ها را از قبل بر روی لپ تاپ مورد نظر نصب کنید و پروژه را بر روی آن تست کنید تا قبل از ارائه از عملکرد صحیح آن اطمینان پیدا کنید.
- جزئیات و زمان ارائه ها متعاقبا خدمت شما اعلام خواهد شد.

#### نکات مهم دیگر:

- قبل از شروع پروژه تیم خود را تشکیل دهید. تیم شما میتواند حداکثر سه عضو داشته باشد و داوری پروژه نیز در حد انتظار از تیم های سه نفره خواهد بود. پس از تشکیل تیم، تقسیم کار مناسب را انجام دهید.
- قبل از شروع پیاده سازی، با توجه به توضیحات پروژه و بر اساس آنچه تا به حال در کلاس خوانده اید، اشیا مورد نیاز در این پروژه را تشخیص داده و نحوه ارتباط هر کدام با یکدیگر را مشخص نمایید. همچنین ویژگی ها و متدهای مربوط به هر کلاس را شناسایی نمایید و دیاگرام کلاس های آن را رسم



نمایید. یکی از قسمت های اصلی نمره تسلط اعضا به دیاگرام و دفاع از دیاگرام تهیه شده خواهد بود.

- علاوه بر موارد ذکر شده در پروپوزال حاضر، می‌توانید با توجه به خلاقیت خود عملکرد های بیشتری را برای برنامه تعریف و آنها را پیاده‌سازی نمایید.
- هر عضو باید وظیفه مشخصی داشته باشد ولی همزمان به بقیه بخش های پروژه نیز تسلط کافی داشته باشد. نمره ارائه هر شخص با توجه به تسلط او بر پروژه از هم‌تیمی های خود متفاوت خواهد بود.
- هر تیم می‌بایست داده های نمونه برای تست برنامه خود آماده کرده باشد تا در روز ارائه برای تست قسمت های مختلف از آن استفاده کند.
- هر تیم می‌بایست علاوه بر فایل های پیاده‌سازی پروژه، گزارشی متنی از پروژه خود تهیه نموده و آن را تحویل دهد. این گزارش بایستی شامل موارد زیر باشد:

- تعریف پروژه و بیان عملکردهای نرم‌افزار پیاده‌سازی شده
- دیاگرام کلاس UML و توضیح هر کلاس به همراه ویژگی ها و متدهای هرکدام

- روش حل مساله و توضیحاتی در پیاده سازی
- نمایش چندین سناریو نمونه برای نمایش نحوه عملکرد نرم افزار
- تیم های خود را حداکثر تا تاریخ 10 خرداد انتخاب کرده و همراه با اسامی تیم به دستیاران آموزشی اعلام نمایید. بعد از تعیین تیم‌ها، امکان تغییر و جابجایی وجود ندارد.
- تاریخ تحویل پروژه متعاقبا خدمت شما اعلام می‌گردد.
- قبل از زمان ارائه لازم است همه اعضای تیم، فایل زیپ پروژه را در قسمت مربوطه در صفحه درس در وبسایت کوئرا بارگزاری نمایند. آپلود فقط تا فرارسیدن تاریخ تحویل پروژه مجاز است و پس از آن امکان بارگزاری به

شکل خودکار غیرفعال می‌گردد و تمدید پروژه پس از اتمام زمان آن امکان‌پذیر نخواهد بود. فایل زیپ شما باید شامل تمام کد های پروژه به همراه مستندات فنی، گزارش پروژه و کلاس دیاگرام UML آن باشد.

- تنها مجاز به استفاده از زبان C++ برای پیاده‌سازی این پروژه خواهید بود.
- در صورت استفاده از پروژه های گیت هاب و ... و یا استفاده از کتابخانه های مختلف خارج از کتابخانه های استاندارد زبان، لازم است ضمن اعلام آن در گزارش پروژه و نیز در زمان ارائه، به دانش استفاده از آنها نیز مسلط باشید. در صورت تشخیص استفاده از سورس های دیگر و اعلام نکردن آن، بر اساس مقدار استفاده در پروژه و عدم دانش شما، پروژه تقلب تشخیص داده خواهد شد و حائز نمره منفی خواهد بود.
- هیچگونه تقلب قابل پذیرش و چشم‌پوشی نخواهد بود و در صورت تشخیص تقلب، مراتب جهت استحضار و پیگیری به دانشکده اعلام خواهد شد. تبعات چنین موضوعاتی متوجه هر دو سمت تقلب دهنده و تقلب گیرنده (فارغ از رشته، سال ورود، اخذ یا عدم اخذ درس و ...) خواهد بود.

موفق و پیروز باشید