

برنامه نویسی پیشرفته

زمستان ۱۴۰۲ و بهار ۱۴۰۳ - دانشکده علوم ریاضی

دانشگاه صنعتی شریف

تمرین سری اول

زمان تحویل: ۱۵ اسفند

در این تمرین می‌خواهیم با [مدل‌سازی](https://en.wikipedia.org/wiki/Object_model) در برنامه‌نویسی و چگونگی ساخت مدل از یک مسئله‌ی واقعی توسط یک برنامه‌ی کامپیوتری آشنا شویم. همچنین در ادامه با پیاده کردن مدل‌هایی که از مسئله مورد نظر ساخته‌ایم با رویکرد شی‌گرا در جاوا آشناتر می شویم.

از آنجایی که شما برای ترم جدید خود انتخاب واحد کردید به این نتیجه رسیدید که سامانه انتخاب واحد را خودتون هم میتونید بزنید، حالا در این تمرین قصد داریم تا ورژن ساده شده انتخاب واحد را با رابط کاربری [CLI](https://en.wikipedia.org/wiki/Command-line_interface) پیاده سازی کنیم. واسط کاربری CLI به این معنا است که ارتباط کاربر با برنامه، از طریق کامند لاین و ترمینال و یا کنسول شکل می‌گیرد. در این حالت کاربر برنامه در کنسول با منطق برنامه ارتباط برقرار می‌کند و دستورات مختلفی که می‌خواهد انجام دهد را به عنوان ورودی در کنسول وارد می‌کند و برنامه شما باید این دستورات را متوجه شود و انجام دهد و نتایج را خروجی دهد. توجه کنید که هدف اصلی این تمرین بیشتر مدل‌سازی و ارث بری  است و CLI برای نشان دادن این مدل‌سازی استفاده می‌شود.

به طور کلی دوست داریم برنامه ای که می‌نویسیم، قابلیت‌های کلی زیر را داشته باشد:

  ما به دو حالت دانشجو و ادمین می‌توانیم وارد شویم.

 دانشجو باید بتواند درس های ارائه شده را با جزئیات آن ها ببیند و بتواند آن ها را در صورت عدم تداخل با درس های فعلی خود، به لیست دروس خود اضافه کند و یا درسی را از لیست دروس خود حذف کند.

ادمین باید بتواند درس‌های ارائه شده هر دانشکده و لیست دانشجو های هر درس را ببیند و بتواند درس های جدید ایجاد کند و همچنین بتواند دانشجویی را از درسی حذف کند و همچنین برای درسی افزایش ظرفیت بزند.

**صفحه ورود:**

در این صفحه که صفحه ابتدایی برنامه شماست باید بتوانیم به دو صورت ادمین یا دانشجو وارد برنامه شویم. همچنین توجه شود برنامه شما باید دارای قابلیتی باشد که در هر زمان بتوانیم به این صفحه برگردیم و عملیات ورود را از ابتدا انجام دهیم.

بدیهی است که در این حالت اگر با کاربری در برنامه بودیم، آن کاربر از برنامه خارج می‌شود زیرا در هر زمان فقط یک کاربر فعال داریم.

نکته: نام کاربری ادمین می‌تواند یک مقدار ثابتی مانند "Admin" باشد زیرا ادمین در این سامانه یکتا است.

**قابلیت های دانشجو:**

1. بتوانیم لیست دروس ثبت نام شده را ببینیم و آنها را حذف کنیم.
2. بتوانیم لیست دروس ارائه شده هر دانشکده را ببینیم و آنها را اخذ کنیم.

در ابتدا هر دانشجو با شماره دانشجویی خود وارد سامانه می‌شود. سپس اگر دانشجو گزینه "لیست دروس ثبت نامی" را انتخاب کرد، باید بتواند تمام دروس ثبت نام شده خود را با جزئیات مانند نام استاد، نام درس، کد درس، ظرفیت، تعداد واحد، زمان کلاس، زمان امتحان، تخصصی یا عمومی بودن درس را ببیند و بتواند یک یا چند درس را با وارد کردن کد درس حذف کند. اگر گزینه "لیست دروس ارائه شده" را انتخاب کرد، لیست تمام دانشکده‌های موجود به دانشجو نشان داده شود و در این مرحله کاربر می‌تواند یکی از دانشکده‌ها را انتخاب کند. اگر دانشجو یکی از دانشکده‌ها را انتخاب کرد، باید بتواند لیست دروس ارائه شده آن دانشکده را با جزئیات مانند نام استاد، نام درس، کد درس، ظرفیت، تعداد واحد، زمان کلاس، زمان امتحان، تخصصی یا عمومی بودن درس ببیند و بتواند یک یا چند درس را با وارد کردن کد درس اخذ کند (در صورت وجود ظرفیت).

**قابلیت های ادمین:**

1. لیست دروس ارائه شده هر دانشکده را ببینیم و درس جدید ایجاد و به لیست دروس ارائه شده آن دانشکده اضافه یا حذف کنیم و یا ظرفیت درسی که قبلا وجود داشته را اضافه کنیم.
2. لیست دانشجو های ثبت نامی هر درس را ببینیم و با استفاده از شماره دانشجویی، دانشجویی را از درس حذف یا  به درس اضافه کنیم.

در ابتدا ادمین با نام کاربری یکتای خود وارد سامانه می‌شود. سپس لیست دانشکده‌های موجود به او نشان داده می‌شود. در این مرحله ادمین می‌تواند یکی از دانشکده‌ها را انتخاب کند. اگر ادمین یکی از دانشکده‌ها را انتخاب کند، باید لیست دروس آن دانشکده به ادمین نشان داده شود و ادمین بتواند درس جدیدی به آن دانشکده اضافه کند یا یکی از دروس فعلی را انتخاب کند و لیست دانشجوهای ثبت نامی آن درس را ببیند و بتواند آن‌ها را با وارد کردن شماره دانشجویی از آن درس حذف کند و همچنین بتواند دانشجویی را با وارد کردن شماره دانشجویی به درس اضافه کند. توجه شود که در زمان اضافه کردن یک درس به دروس ارائه شده یک دانشکده، باید تمام جزئیات آن درس اعم از نام درس، نام استاد، کد درس، ظرفیت، تعداد واحد، زمان امتحان، زمان کلاس، تخصصی یا عمومی بودن درس توسط ادمین ثبت شود. امکان اضافه کردن ظرفیت هر درس نیز باید وجود داشته باشد.

**قوانین انتخاب واحد:**

1. حداکثر واحد قابل اخذ برای هر دانشجو 20 است و هیچ دانشجویی نباید بتواند بیشتر از این مقدار واحد اخذ کند.
2. در زمان اخذ باید لحاظ شود که درس انتخابی با دروس ثبت نام شده تداخل نداشته باشد (زمان امتحان و کلاس) در صورت وجود تداخل دانشجو قادر به اخذ درس نبوده.
3. در صورت پر بودن ظرفیت یک درس، دانشجو نمی‌تواند آن درس را اخذ کند.
4. هر دانشجو حداکثر 5 واحد عمومی میتواند انتخاب کند.

**نکات تکمیلی:**

* کلاس درس (course) باید در برنامه شما وجود داشته باشد و به دو زیر کلاس **عمومی** و **تخصصی** تقسیم شود و استفاده از مباحث ارث بری الزامی است و جزئی از ارزشیابی می‌باشد.
* در مدل سازی شما  دانشگاه باید حداقل ۴ دانشکده و هر دانشکده حداقل 3 درس ارائه شده داشته باشد (هر دانشکده باید حداقل یک درس عمومی ارائه دهد). این اطلاعات باید به صورت Hardcode  در\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* سامانه شما وارد شوند یعنی قبل از اجرا برنامه این اطلاعات در برنامه وجود داشته باشند.
* ممکن است یک سری خطای منطقی در برنامه ما رخ دهد که ما باید از آنها جلوگیری کنیم. به این کار مدیریت خطا( Error Handling) گفته میشود. به عنوان مثال اگر در بخش انتخاب واحد، برای اخذ درس کد وارد شده متعلق به هیچ کدام از دروس موجود نباشد باید با چاپ پیام مناسبی در کنسول به کاربر نشان دهید که کد درس وارد شده درست نیست.
* در هر قسمتی، ما همیشه باید یک مکانیزم برای کنسل کردن یک عملیات (مثلا کنسل کردن عمل افزایش ظرفیت درسی قبل از وارد شدن مقدار افزایش) یا برگشت به عقب (مثلا از قسمت منوی دانشجو به صفحه ورود) داشته باشیم. مثلا هر وقت کاربر کلمه ی back را در کنسول نوشت، برنامه ما یک مرحله به عقب برگردد و به صفحه قبلی بازگردد.

* توصیه می‌شود که سعی کنید ابتدا درباره ساختار و مدلی که دوست دارید این برنامه را پیاده‌سازی کنید خوب فکر کنید و تلاش کنید که تا حد امکان موارد مختلف برنامه (مثلا مدل‌ها، منطق برنامه، رابط کاربری CLI و …)  را تا حد خوبی از یکدیگر جدا کنید.

این کار به شما کمک می‌کند که بتوانید با یک ساختار ذهنی منظم به سراغ این تمرین بروید.

* برنامه شما باید رابط کاربری خوبی داشته باشد. یعنی اگر یک فرد غریبه کد شما را اجرا کند، به راحتی بتواند دستوراتی که برنامه شما از آنها پشتیبانی کند را اجرا کند و اگر در مرحله‎‌ای موفق به انجام کاری نشد دلیل آن را ببیند (به عنوان مثال اگر درسی که دانشجو می‌خواهد اخذ کند ظرفیت نداشته باشد باید به کاربر اعلام شود که به دلیل عدم وجود ظرفیت اخذ درس امکان پذیر نیست). برای این منظور، برنامه شما باید در هر مرحله، کارهایی که در هر مرحله کاربر میتواند انجام دهد را به طریقی به کاربر نشان دهد.
* استفاده از ابزارهای version control مثل Git **اجباری** است. استفاده صحیح از Git برای تمرین دارای نمره خواهد بود. همچنین باید موارد زیر را رعایت کنید:

1. یک ریپازیتوری پرایوت داشته باشید و بعد از پایان تمرین نسخه نهایی را پوش کنید.
2. با توجه به مسیری که در حل تمرین طی می‌کنید Branch های مختلف بسازید.
3. حتماً کامیت‌های منظم داشته باشید و توضیحات کامیت را بنویسید.

**بخش امتیازی (اختیاری)**

**احراز هویت:**

در بخش **اجباری** این تمرین، یک کاربر برای ورود به سامانه صرفا با وارد کردن شماره دانشجویی یا نام کاربری ادمین وارد سامانه می‌شد؛ در این بخش **اختیاری** ما از شما می‌خواهیم که قابلیت **احراز هویت** را به سامانه اضافه کنید.

یعنی در صفحه ورود، کاربر علاوه بر قابلیت ورود به حساب کاربری (log in)، قابلیت ساخت اکانت جدید (sign up) را داشته باشد. و با وارد کردن نام کاربری و پسورد، اکانت خود را بسازد.

توجه شود در این صورت کاربر برای ورود به اکانت خود (log in) باید علاوه بر نام کاربری، رمز اکانت که در موقع ساخت اکانت تعیین کرده بود را نیز وارد کند و سامانه شما باید بتواند صحت رمز و نام کاربری وارد شده را بسنجد و در صورت **مغایرت**، اجازه ورود به سامانه را به کاربر **ندهد**.

**نکته:** رمز اکانت ادمین قابل تغییر نیست و ثابت است و همچنین باید قبل از شروع اجرا برنامه در سامانه ذخیره شود.

**کار با فایل:**

به دلیل علاقه‌ی شما به کار با فایل‌ها، در این بخش قصد داریم یک سیستم ساده برای ذخیره و بارگیری داده‌ها برای ادمین و دانشجو ایجاد کنیم. این سیستم به ادمین امکان می‌دهد تا درس (course) هایی که اضافه می‌کند را ذخیره کرده و در آینده مجدداً بارگیری کند.

برای این بخش، باید یک پنل ادمین در رابط کاربری (CLI) ایجاد کنیم که دارای یک گزینه به نام "export" و یک گزینه به نام "import" باشد.

با انتخاب گزینه "export"، از ادمین باید درخواست شود تا آدرس یک فایل را وارد کند و تمامی درس هایی که اضافه کرده است در آن فایل ذخیره شوند. همچنین، گزینه "import" باید قادر باشد آدرس یک فایل را بپذیرد و تمامی درس ها را از آن فایل بخواند و به اضافه کند. و اما برای دانشجو باید تمامی دروس اخذ شده هر دانشجو را بصورت خودکار ذخیره کرده تا در استفاده های بعدی اگر با همان دانشجو وارد سامانه شدیم دروس ثبت نامی اخذ شده از قبل را ببینیم.  
همچنین باید اطلاعاتی که بصورت Hardcode در کد خود قرار داده‌اید را با کمک فایل خوانده و در برنامه خود استفاده کنید. سیستم احراز هویت نیز باید بوسیله فایل صورت بگیرد یعنی یوزر های ساخته به‌همراه رمز باید ذخیره شوند تا در استفاده های دیگر بتوانیم وارد یوزر های ساخته شده قبل شویم.