

به نام خدا

## تعریف فاز سوم پروژه میان‌ترم درس برنامه‌نویسی پیشرفته

۰. فایل مربوط به توضیحات نحوه ارسال تمرین‌ها را که در مدل قرار دارد، مطالعه کنید.

۱. تمامی فایل‌های کد و مستندات را به صورت یک فایل آرشیو zip (zip != rar) که به قالب زیر نام‌گذاری شده باشد، بارگذاری نمایید. دقت کنید که هر فاز تحویل حضور جداگانه دارد.

StudentNumber\_FirstName\_LastName.zip

9031806\_Mohammad\_Ahmadpanah.zip

۲. مستندسازی به کمک JavaDoc، کامنت‌گذاری و رعایت اصول کدنویسی خوانا برای همه کلاس‌های پیاده‌سازی شده الزامی است.

۳. طراحی مناسب و رعایت اصول شی‌گرایی الزامی است. برنامه شما باید آمادگی اعمال تغییرات بعدی در نیازمندی‌های پروژه را داشته باشد.

۴. استفاده از مخزن خصوصی (private repository) گیت‌لب (GitLab) برای کنترل نسخه‌های برنامه الزامی است. شما باید دسترسی خواندن مخزن مربوط به پروژه را در اختیار مدرس کارگاه خود قرار دهید. توضیحات بیشتر در ویدیوی آموزشی موجود در مدل قرار دارد و می‌توانید از مدرس کارگاه خود کمک بگیرید. باید commit‌های مناسب و منطقی در نسخه‌های مختلف برنامه شما وجود داشته باشد.

۵. سوالات خود را درباره این پروژه می‌توانید از طریق فروم موجود در مدل بپرسید. مطالب گفته‌شده در آن فروم توسط مدرسان درس یا کارگاه، بخشی از تعریف پروژه محسوب می‌شود.

**مهلت تحویل: تا شنبه ۱۲ خرداد ۱۳۹۷ ساعت ۷:۰۰ صبح**

## طراحی و پیاده‌سازی نرم‌افزار مدیریت دانلود JDM

هدف از این پروژه، طراحی و پیاده‌سازی نرم‌افزاری برای مدیریت دانلود به زبان جاوا است. طراحی و پیاده‌سازی قسمت‌های مختلف این پروژه را در قالب فازهایی به شما محول کرده‌ایم و در هر فاز، قسمت‌هایی از نیازمندی‌های نرم‌افزار را مشخص می‌کنیم. بنابراین، در این فازها، نیازمندی‌های مختلف نرم‌افزار به شما معرفی می‌شود و باید در هر فاز، موارد خواسته‌شده به طور دقیق و کامل پیاده شوند. فاز سوم (آخر) به طراحی و پیاده‌سازی **بخش چندریسگی و شبکه** این نرم‌افزار اختصاص دارد. دقت کنید که باید همه قسمت‌های پیاده‌سازی‌شده در فازهای قبلی نیز به درستی کار کنند و با توجه به محتوای این فاز، به‌روز شوند تا یک برنامه کامل مدیریت دانلود تحویل داده شود.

## طراحی و پیاده‌سازی بخش چندریسی و شبکه

### نرم‌افزار مدیریت دانلود JDM

در فاز اول، رابط کاربری گرافیکی به همراه بخشی از کارکرد برنامه را پیاده‌سازی کرده‌اید. در فاز دوم، قسمت‌های مربوط به کار با فایل و رشته‌های متنی مطرح شده بود. در این فاز، نیازمندی‌های پروژه مربوط به چندریسی و شبکه مشخص می‌شوند. در ادامه، باید موارد خواسته‌شده را به عملکرد برنامه اضافه کنید و در صورت نیاز، رابط کاربری گرافیکی متناسب با آن را طراحی و پیاده‌سازی کنید. همان‌طور که در توضیح فازهای قبل عنوان شد، هدف این نیست که تمامی امکانات نرم‌افزار EagleGet توسط شما پیاده‌سازی شود، اما از نظر ظاهری و رابط کاربری گرافیکی باید طراحی شما شبیه این نرم‌افزار باشد. در صورتی که نیاز است تا رابط کاربری گرافیکی جدیدی طراحی شود، باید سازگار با رابط کاربری این نرم‌افزار و کاربرپسند باشد. تمامی قسمت‌های فازهای قبل باید با توجه به این فاز تکمیل شوند و همه اجزای برنامه به درستی عمل کند. پاسخ‌دهی مناسب و سریع برنامه نسبت به درخواست‌های کاربر از برنامه اهمیت دارد.

نیازمندی اصلی این فاز، همان نیازمندی اصلی نرم‌افزار یعنی دانلود فایل‌ها به طور همزمان است. جزئیات نیازمندی‌های این فاز عبارتند از:

- در صورتی که دانلود جدیدی به لیست دانلودها اضافه شود، باید این دانلود همزمان با سایر دانلودهای فعال در لیست دانلودها پیشرفت داشته باشد. لازم به ذکر است که دانلودهای موجود در صف، این‌گونه نیستند؛ بلکه باید ابتدا دانلودهای قبلی کاملاً پایان یابد و سپس دانلود فعلی آغاز شود. تمامی دکمه‌های موجود در نوار ابزار باید کارکرد خود را در تغییر وضعیت یک دانلود داشته باشند. نوار پیشرفت هر دانلود باید بر اساس سرعت دانلود و نسبت مقدار دانلودشده به حجم کل فایل (درصد دانلود) پیشروی داشته باشد.
- در صورتی که در تنظیمات برنامه تعداد دانلودهای همزمان محدود شده باشد، باید این محدودیت در کارکرد برنامه مشخص باشد. به این معنا که تنها به تعداد مشخص‌شده دانلودهای همزمان صورت بگیرند و سایر دانلودها منتظر پایان یافتن دانلودهای قبلی باشند. به محض اینکه یک دانلود پایان یافت، دانلودی که منتظر بوده است، شروع به کار می‌کند.
- تنها پس از دانلود کامل و موفقیت‌آمیز یک فایل، با دوبار کلیک کردن روی نام فایل باید آن فایل به درستی باز شود.

- دقت شود که در تنظیمات برنامه، لیست سایت‌های غیرمجاز برای دانلود مشخص شده است. بدیهی است که امکان دانلود فایل‌ها برای آدرس‌های موجود در آن لیست نباید وجود داشته باشد و این نکته با پیامی به کاربر اطلاع داده شود. کاربر می‌تواند لیست سایت‌های غیرمجاز را در برنامه ویرایش کند و مجدداً دانلود فایل را درخواست کند.
- عملیات جستجو و مرتب‌سازی لیست دانلودها باید برای همه دانلودها (به غیر از دانلودهای حذف‌شده) به درستی انجام شود. لیست دانلودها باید بر اساس درخواست جستجو یا مرتب‌سازی به‌روز شود.
- در صورتی که برنامه زمان‌بندی برای دانلودی تعیین شده باشد، باید با در نظر داشتن محدودیت تعداد دانلود همزمان، دانلود فایل طبق زمان‌بندی انجام شود. در صورتی که پیش از رسیدن به زمان مقرر، برنامه کاملاً بسته شود و پس از بازشدن مجدد برنامه، هنوز فرصت باقی‌مانده باشد، دانلود از زمان مقرر آغاز می‌شود. در غیر این صورت، پیامی مبنی بر انقضای زمان‌بندی به کاربر نمایش داده شود.

در ادامه توضیحاتی در خصوص نحوه دانلودکردن فایل‌ها از آدرس‌های لینک دانلود آمده است:

برای انجام دانلود از یک وب‌سایت می‌توان از پروتکل HTTP استفاده کرد. پروتکل HTTP از بستر TCP جهت انتقال داده استفاده می‌کند. این پروتکل در ساده‌ترین حالت، به صورت تقاضا و پاسخ عمل می‌کند؛ به این ترتیب که به سرور تقاضای دریافت یک فایل را می‌دهید و سرور در جواب این تقاضا، پاسخی را ارسال می‌کند. ساده‌ترین نوع تقاضا در این پروتکل، تقاضای GET است که ساختار آن به صورت زیر است:

```
GET /hello.zip HTTP/1.1  
Host: aut.ac.ir
```

همانطور که مشاهده می‌کنید، در این پیام می‌بایست دو مقدار Host و نام فایل توسط برنامه تعیین شوند. به طور مثال، اگر بخواهیم از آدرس `http://www.aut.ac.ir/hello.zip` فایل `hello.zip` را دانلود کنیم، پیامی با محتوای بالا به سرور ارسال می‌کنیم.

توجه کنید که یک آدرس لینک دانلود (یا به بیان دقیق‌تر، یک URL) از دو قسمت تشکیل می‌شود. قسمت اول که تا اولین کاراکتر / است، Host را مشخص می‌کند که ارتباط TCP با همین Host برقرار

می‌گردد و از خود کاراکتر / به بعد، آدرسی را مشخص می‌کند که شما برای آن تقاضا دارید. دقت کنید که این آدرس در پیام تقاضا، بعد از واژه GET نوشته شده است. پروتکل HTTP برای مشخص کردن وضعیت و دسته‌بندی انواع پاسخ‌ها، از یک کد سه رقمی استفاده می‌کند. اگر این کد در پیام پاسخ به یک تقاضا برابر با ۲۰۰ باشد، به این معناست که تقاضای شما با موفقیت پردازش شده است.

ساده‌ترین پاسخ در HTTP که حامل داده نیز هست، در قالب زیر قرار می‌گیرد:

```
HTTP/1.1 200 OK
...
Content-Length: 100
...

[100 bytes of data]
```

اما آنچه قرار است برای بخش شبکه برنامه خود انجام دهید این است که کاربر بتواند با وارد کردن URL با پروتکل‌های HTTP و HTTPS فایل درخواستی خود را از سرور دریافت و دانلود کند. این کار را با استفاده از یکی از کتابخانه‌های آماده HTTP و HTTPS در جاوا انجام دهید (راهنمایی: HttpURLConnection). دانلود فایل از لینک‌های مستقیم (که آدرس فایل موجود در سرور از طریق URL دقیقاً مشخص شده است) کفایت می‌کند.

با توجه به این‌که برنامه شما باید امکان دانلود همزمان را داشته باشد، می‌بایست به ازای هر دانلود کاربر یک ریس (thread) ساخته و در آن تقاضای دریافت فایل را بفرستید. برای این کار در کتابخانه‌های آماده، متدهای مشخصی وجود دارد و باید از طریق مستندات کتابخانه با آن‌ها آشنا شوید. دقت کنید که در صورتی که سرور تقاضای شما را بپذیرد، پاسخی با کد ۲۰۰ همراه با محتوای فایل درخواستی ارسال می‌کند. سپس ریس مربوط به آن دانلود نیز محتوای دریافتی را خوانده و آن را در یک فایل هم‌نام با فایل درخواستی ذخیره می‌کند. در غیر این صورت، دانلود کاربر ناموفق تلقی می‌شود و از طریق پیامی به کاربر اطلاع داده می‌شود.

همواره دانلودها را در ریس‌های مجزا اجرا کنید؛ زیرا این کار طولانی می‌تواند کار سایر قسمت‌های برنامه را مختل کند. پس در کل، برنامه شما یک ریس برای مدیریت رابط کاربری گرافیکی و تعدادی ریس برای دانلودهایی دارد که هم اکنون در حال انجام هستند.

امتیازی: امکان دانلود برای لینک‌های غیرمستقیم نیز وجود داشته باشد.

امتیازی: برای افزایش سرعت در دانلود فایل، از تکنیک دانلود multi-segment (چندپاره) استفاده کنید.