



مهندسی نرم افزار ۱



تکلیف صفر
آشنایی با جاوا

مدرس: دکتر فاطمه قاسمی

رشته: مهندسی کامپیوتر

دستیار آموزشی: **عرفان احمدی**

نیمسال دوم سال تحصیلی 1403-04

هدف پروژه:

هدف این تکلیف مرور ویژگی‌هایی از زبان جاوا است که ممکن است کمتر با آن آشنا باشید اما در جریان انجام تکلیف‌های بعدی به آنها نیاز پیدا می‌کنید. این تکلیف در چند گام تعریف شده و در هر گام به بخش‌هایی از مراجع آموزش جاوا ارجاع داده شده است که مطالعه آنها توصیه می‌شود. در خیلی از موارد به بخشی‌هایی از کتاب Core Java for the Impatient ارجاع داده شده که با کوتاه‌نوشت CJI به آن اشاره شده است.

گام ۱ - نصب جاوا

در صورتی که روی سیستم خود ابزارهای توسعه جاوا (JDK) نصب نکرده‌اید این کار را انجام دهید. برای این که مشکلات تحریم دردرس کمتری ایجاد کند توصیه می‌شود از [openjdk](#) استفاده کنید که نصب آن با apt قابل انجام است. برای انجام پروژه‌های درس توصیه می‌شود نسخه ۲۱ یا بالاتر را نصب کنید.

گام ۲ - ایجاد پروژه

در این گام یک پروژه ایجاد می‌کنیم که در گام‌های بعد آن را کامل می‌کنیم. از آنجا که در گام‌های بعد از کتابخانه CSVReader استفاده خواهیم کرد، این پروژه باید با ابزار میون (Maven) ساخته شود. برای ایجاد پروژه می‌توانید [مستقیماً از میون استفاده کنید](#) یا برای راحتی کار می‌توانید از [محیط برنامه‌نویسی آیدیا](#) (IntelliJ IDEA) استفاده کنید. نسخه رایگان Community Edition این ابزار برای انجام پروژه‌های این درس کافی است. پس از نصب آیدیا، یک پروژه جدید ایجاد کنید و Build System آن را Maven تعیین کنید. نیازی به ایجاد مخزن گیت برای پروژه نیست و برای راحتی کار می‌توانید Add sample code را تیک بزنید تا کد نمونه برایتان تولید شود.

گام ۳ - مطالعه کلاس Date

در پروژه ساخته شده، یک کلاس Date ایجاد کنید و بدنه آن را از روی [پیاده‌سازی داده‌شده](#) کپی کنید. این پیاده‌سازی را مطالعه کنید و مطمئن شوید نکات زیر را می‌دانید:

- اشیاء در جاوا همیشه از طریق ارجاع (مشابه اشاره‌گر در C++) مورد استفاده قرار می‌گیرند [- CJI Section 2.1.2]
 - چرا فیلدهای این کلاس به صورت final تعریف شده‌اند [CJI - Section 2.3.6]
 - سازوکار استثنای در جاوا چگونه است [CJI - Section 5.1]
 - کلاس Object در کتابخانه استاندارد چیست؟ سه متد equals و hashCode و toString چه نقشی دارند؟ [CJI - Section 4.3]
- بدنه متد main کلاس Main را تغییر دهید: یک شیء تاریخ بسازید که معادل ۱۴۰۲/۱۱/۱۸ باشد. فردای این روز را در خروجی بنویسید.

گام ۴ - اضافه کردن مقایسه به کلاس Date

ابتدا درباره رابط‌ها (interfaces) در جاوا مطالعه کنید. برای یک شروع نسبتاً ساده می‌توانید از [این خودآموز](#) استفاده کنید. [CJI - Section 3.1] هم منبع خوبی برای مطالعه است. درباره رابط Comparable مطالعه نمایید [CJI - Section 3.3] و کلاس Date را Comparable نمایید.

گام ۵ - اضافه کردن کلاس‌های Actor و Starring

ابتدا یک کلاس به نام Starring تعریف کنید که مدل‌کننده نقش‌آفرینی در یک فیلم در یک بازه زمانی است. برای آن سه عضو تعریف کنید: نام فیلم از نوع رشته، تاریخ شروع و پایان فیلمبرداری از نوع تاریخ (کلاس Date). کنترل‌های لازمه را در سازنده انجام دهید و در صورتی که آرگومان‌های آن نامعتبر باشند IllegalArgumentException پرتاب کنید. همچنین متدی تعریف کنید که یک بازه دیگر را می‌گیرد و بسته به این که دو بازه هم‌پوشانی (حتی در یک روز) داشته باشند مقدار درست یا نادرست برگرداند. دو بازه [b1, e1] و [b2, e2] هم‌پوشانی دارند اگر (e1 >= b2) && (b1 <= e2). متد دیگری برای این کلاس تعریف کنید تا تعداد روزهای یک عضویت را برمی‌گرداند.

سپس یک کلاس به نام Actor تعریف کنید که علاوه بر نام بازیگر، تاریخچه نقش‌آفرینی او در فیلم‌های مختلف را در قالب یک لیست نگه می‌دارد. برای تعریف این لیست از ArrayList استفاده کنید که شبیه به vector در C++

است. یک متد برای Actor تعریف کنید که بتوان با آن یک سابقه نقش‌آفرینی به آن بازیگر اضافه کرد. برای این متد عدم هم‌پوشانی نقش‌آفرینی جدید با نقش‌آفرینی‌های دیگر بازیگر را کنترل کنید.

گام ۶ - محاسبات تابعی

متدی برای بازیگر تعریف کنید که نام یک فیلم را می‌گیرد و تعداد کل روزهایی را که بازیگر در حال بازی کردن در آن فیلم بوده را برمی‌گرداند. برای پیاده‌سازی این متد از سبک محاسبات تابعی (functional) استفاده کنید. به این منظور با امکانات stream در جاوا آشنا شوید ([خودآموز مقدماتی](#)، [CJL - Sections 8.1-8.3]) و از عبارتهای لامبدا [CJL - Section 3.4] استفاده کنید. به این ترتیب بدنه این متد در یک خط قابل پیاده‌سازی است.

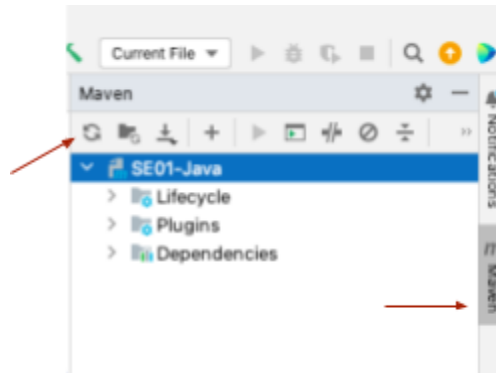
گام ۷ - خواندن سوابق از فایل

در این گام متد main برنامه را کامل می‌کنیم تا اطلاعات سوابق بازیگر را از یک فایل سی‌اس‌وی بخواند و محاسباتی را روی آنها انجام دهد. به این منظور، از کتابخانه [OpenCSV](#) است استفاده می‌کنیم. پیشنهاد می‌شود برای انجام این گام از [CSVReader](#) استفاده کنید.

برای اضافه کردن کتابخانه OpenCSV به پروژه، لازم است pom.xml را که ساختار پروژه شما و وابستگی‌های آن به کتابخانه‌های خارجی را برای میون توصیف می‌کند تغییر دهیم. به این منظور این فایل را در پروژه خود پیدا کنید (در همان پوشه پروژه) و آن را در آیدیا باز کنید. در آخرین قسمت عنصر project (بعد از properties) باید بخش زیر را کپی نمایید:

```
<dependencies>
  <dependency>
    <groupId>com.opencsv</groupId>
    <artifactId>opencsv</artifactId>
    <version>5.8</version>
  </dependency>
</dependencies>
```

(اگر احیاناً pom.xml شما از قبل عنصر dependencies دارد، فقط dependency فوق را داخل dependencies موجود کپی کنید). ممکن است بعد از انجام این تغییر شماره نسخه (۵.۸) به رنگ قرمز در بیاید. در این صورت با استفاده از امکان Reload All Maven Projects (شکل زیر) این مشکل را حل کنید).



پس از اضافه کردن وابستگی فوق در pom.xml می‌توانید از کلاس‌های کتابخانه مذکور (مثل CSVReader) استفاده کنید. برای حل این قسمت می‌توانید کد خود را برای خوانایی بیشتر در قالب متدهایی استاتیک در کلاس Main سازماندهی کنید.

فرض کنید اطلاعات سوابق بازیگران در یک فایل به نام input.csv در پوشه پروژه شما موجود است. محتوای این فایل می‌تواند شبیه به این باشد:

```
James Stewart,Rear Window,1,3,1953,14,5,1953
James Stewart,The Man Who Knew Too Much,13,6,1955,14,9,1955
James Stewart,Vertigo,1,4,1957,31,5,1957
Jack Lemon,Some Like It Hot,7,9,1958,29,12,1958
Jack Lemon,The Apartment,1,1,1959,29,12,1959
```

در نهایت، با استفاده از مثال فوق، محاسبه کنید جیمز استوارت چند روز در حال بازی کردن در فیلم‌های مختلف بوده است.

نحوه تحویل

در پاسخ تکلیف پوشه پروژه خود را (به جز زیرپوشه target) در قالب یک فایل زیپ به نام `A0-{StudentNo}.zip` فشرده کنید و در محل مشخص شده در ای لرن بارگذاری نمایید (به جای {StudentNo} شماره دانشجویی خود را قرار دهید).

در صورت هرگونه مشکل یا ابهام می‌توانید از طریق [ایمیل](#) یا [تلگرام](#) با من در ارتباط باشید.

موفق باشید!