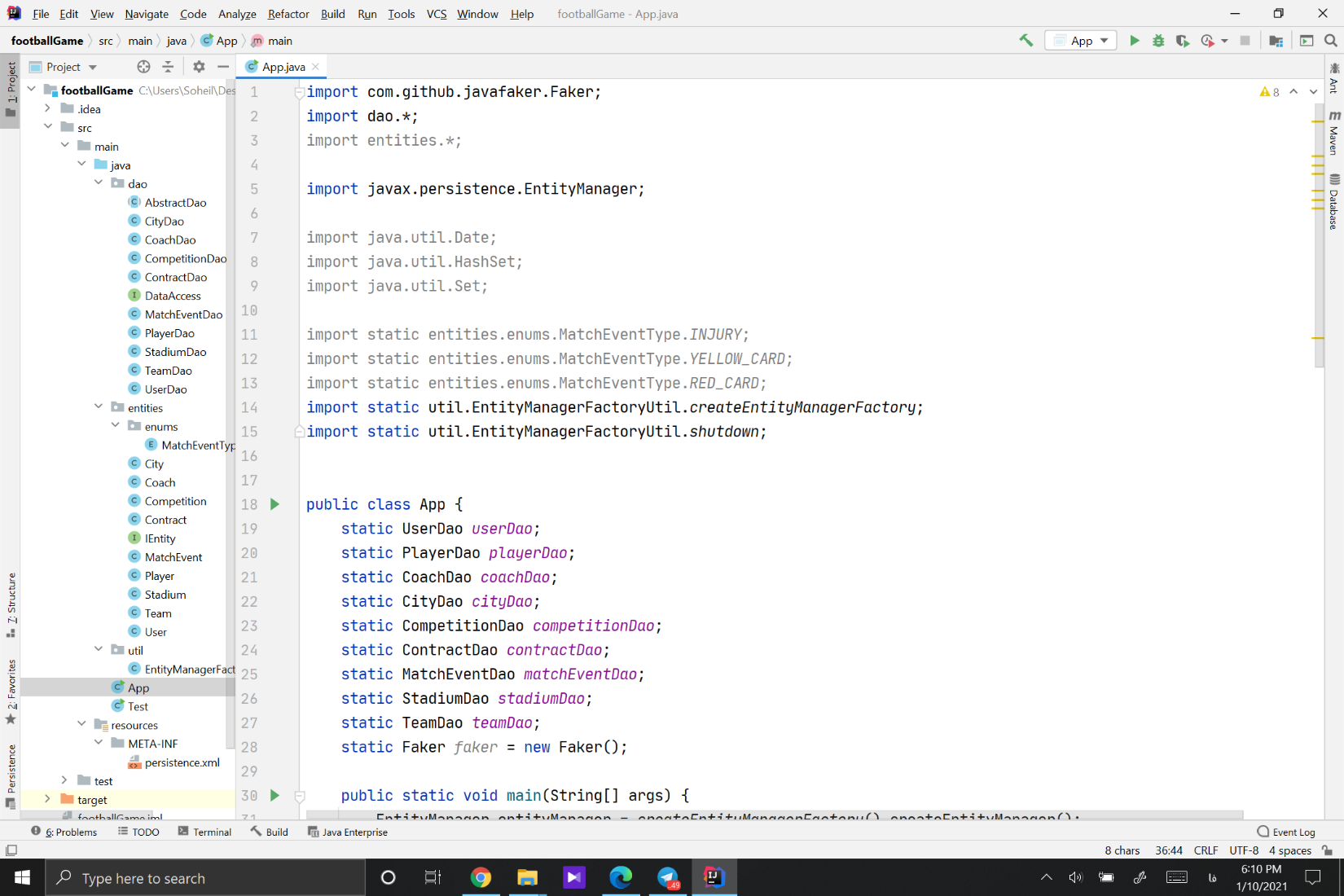
**به نام خداوند بخشنده مهربان**

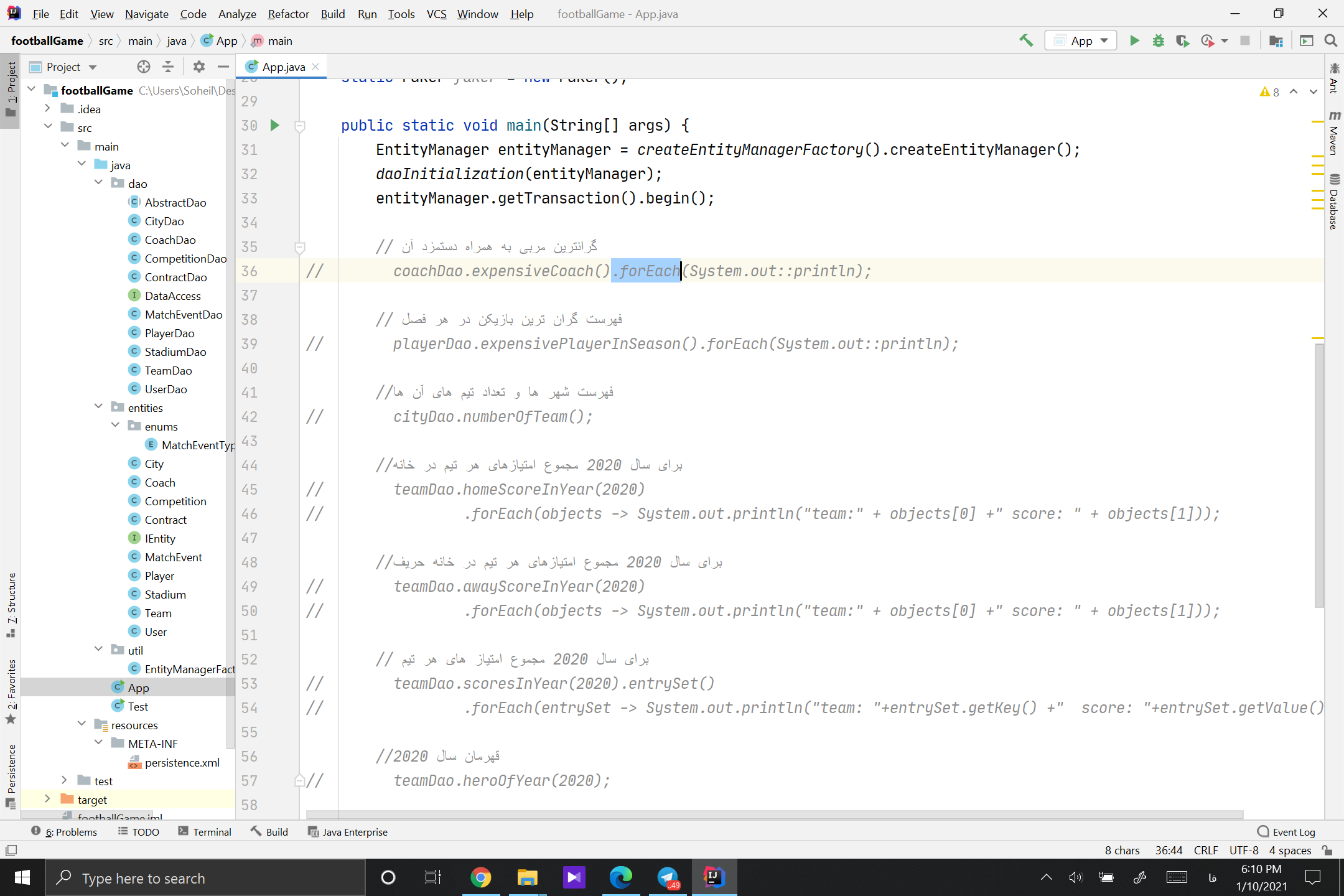
**محمدرضا مسیب زاده**

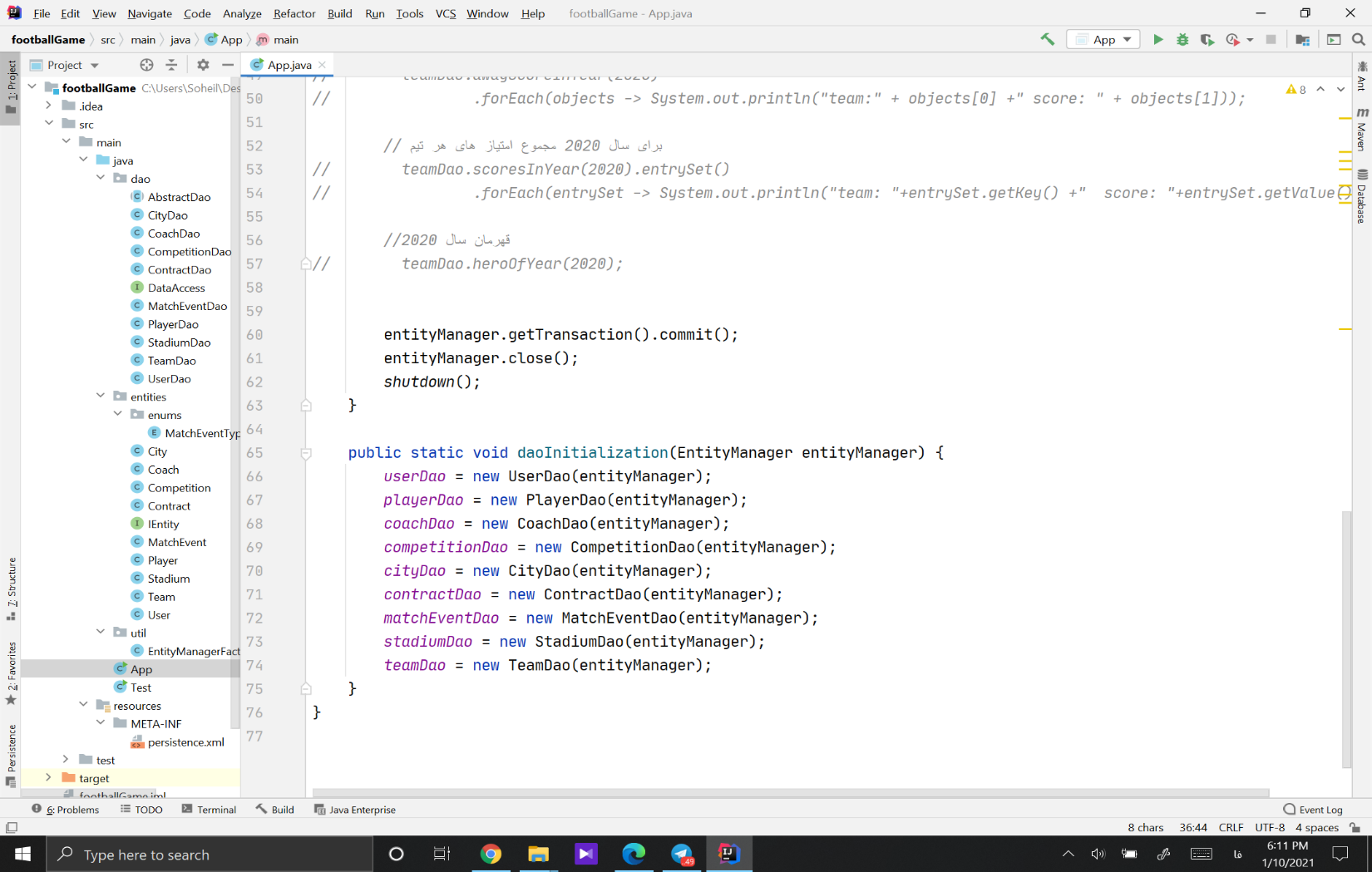
تمرین جاوا :

https://github.com/M-Mosaiebzadeh/Maktab\_W17.git

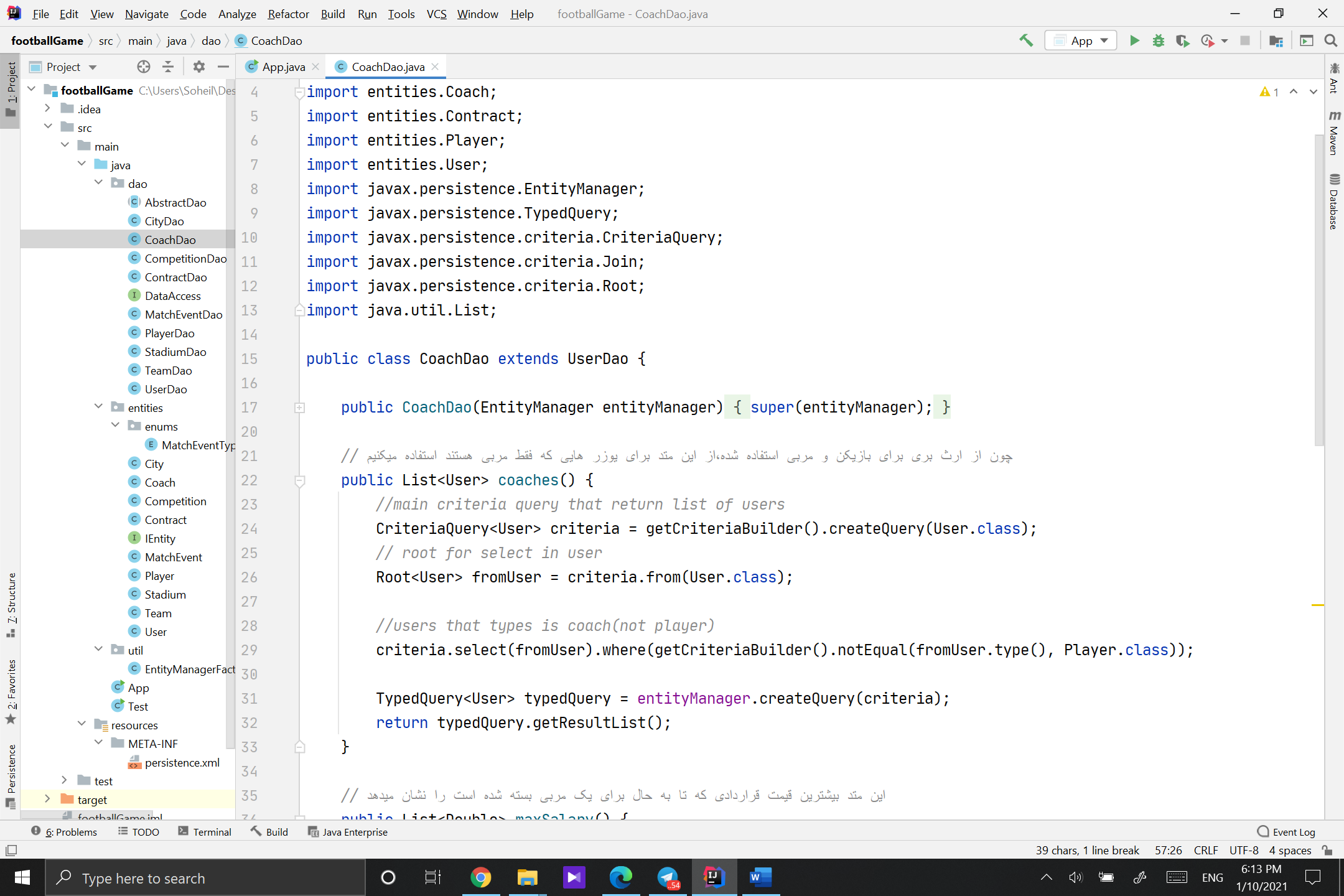
تمرین شماره 1:

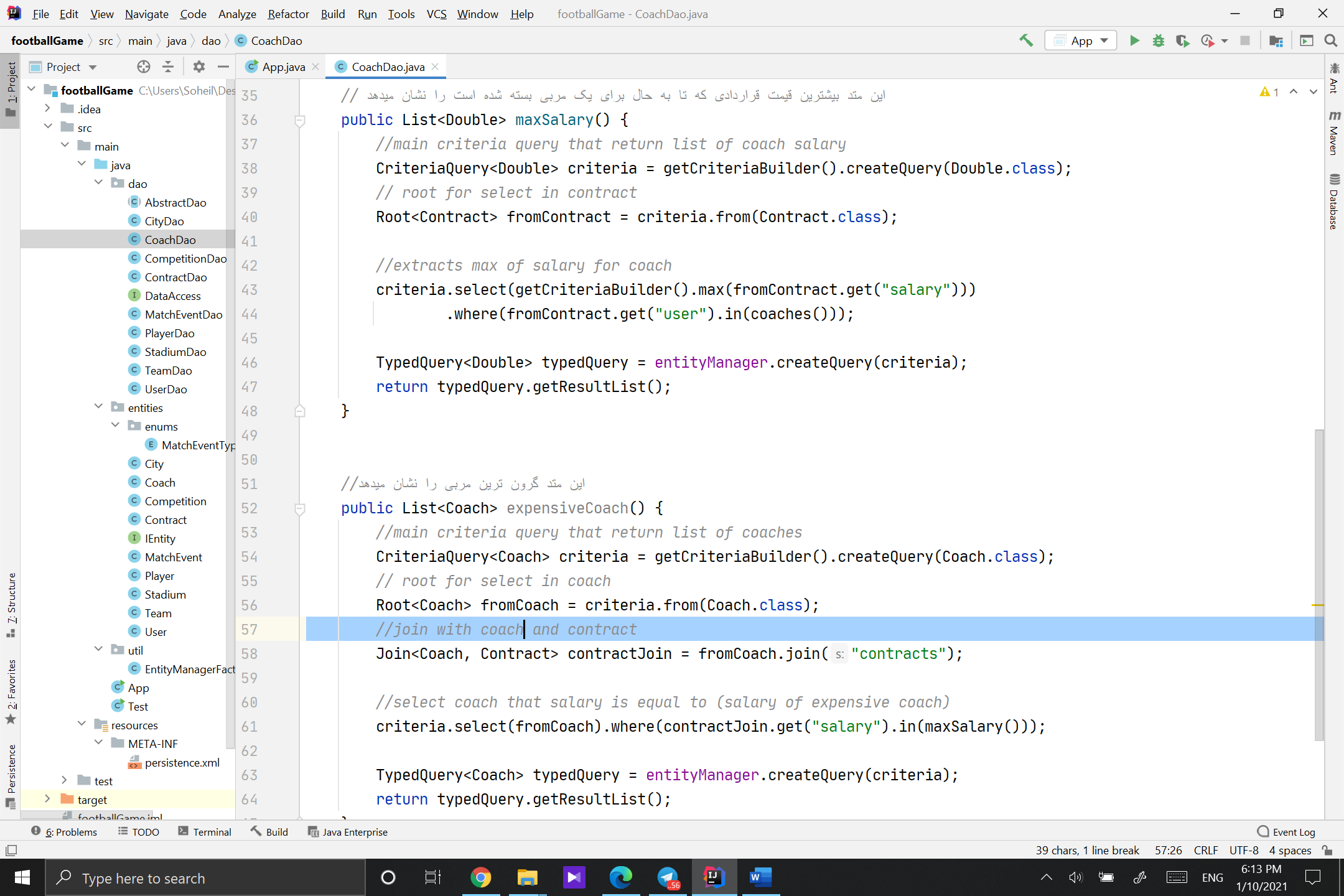






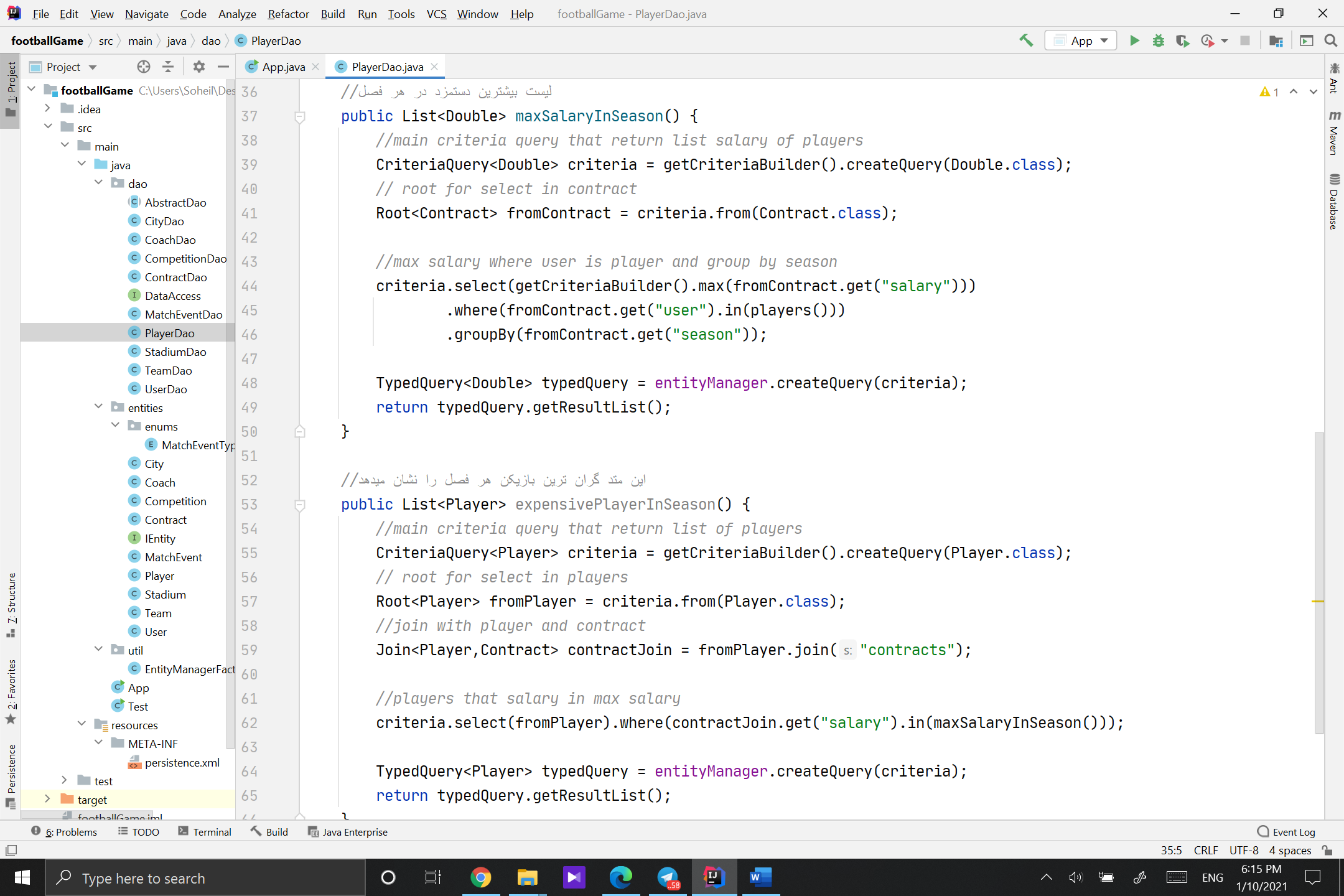
coachDao:



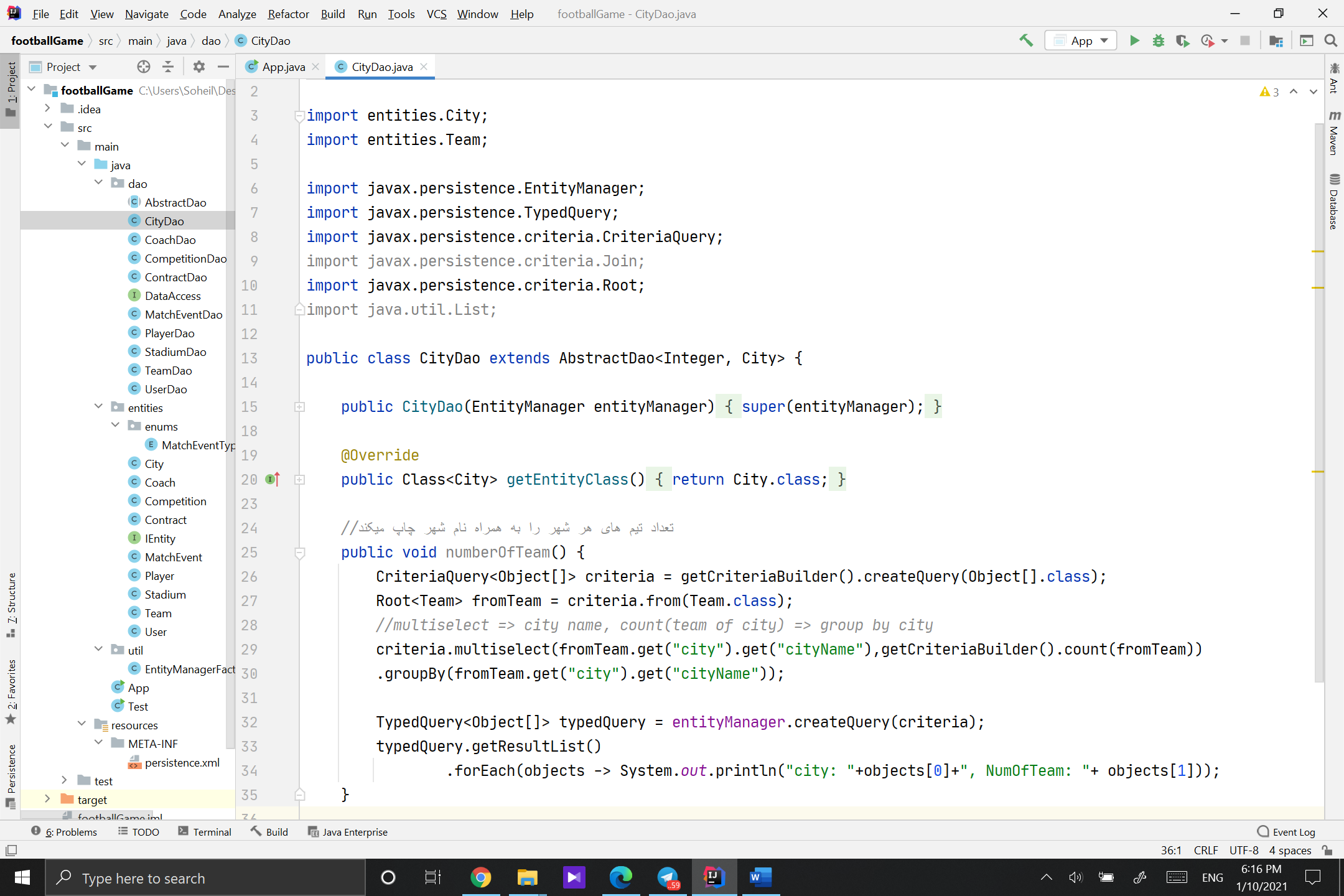


playerDao:

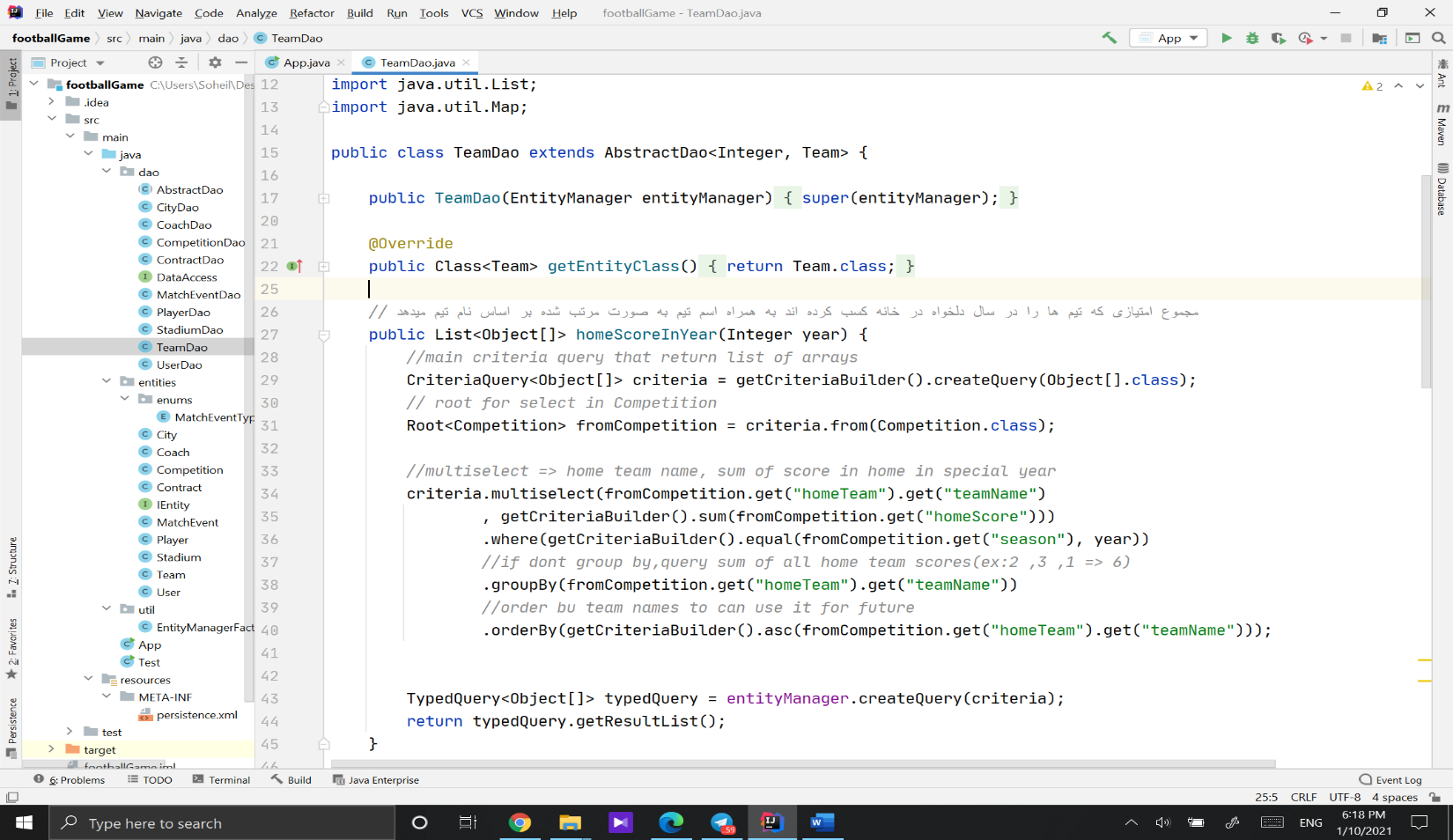


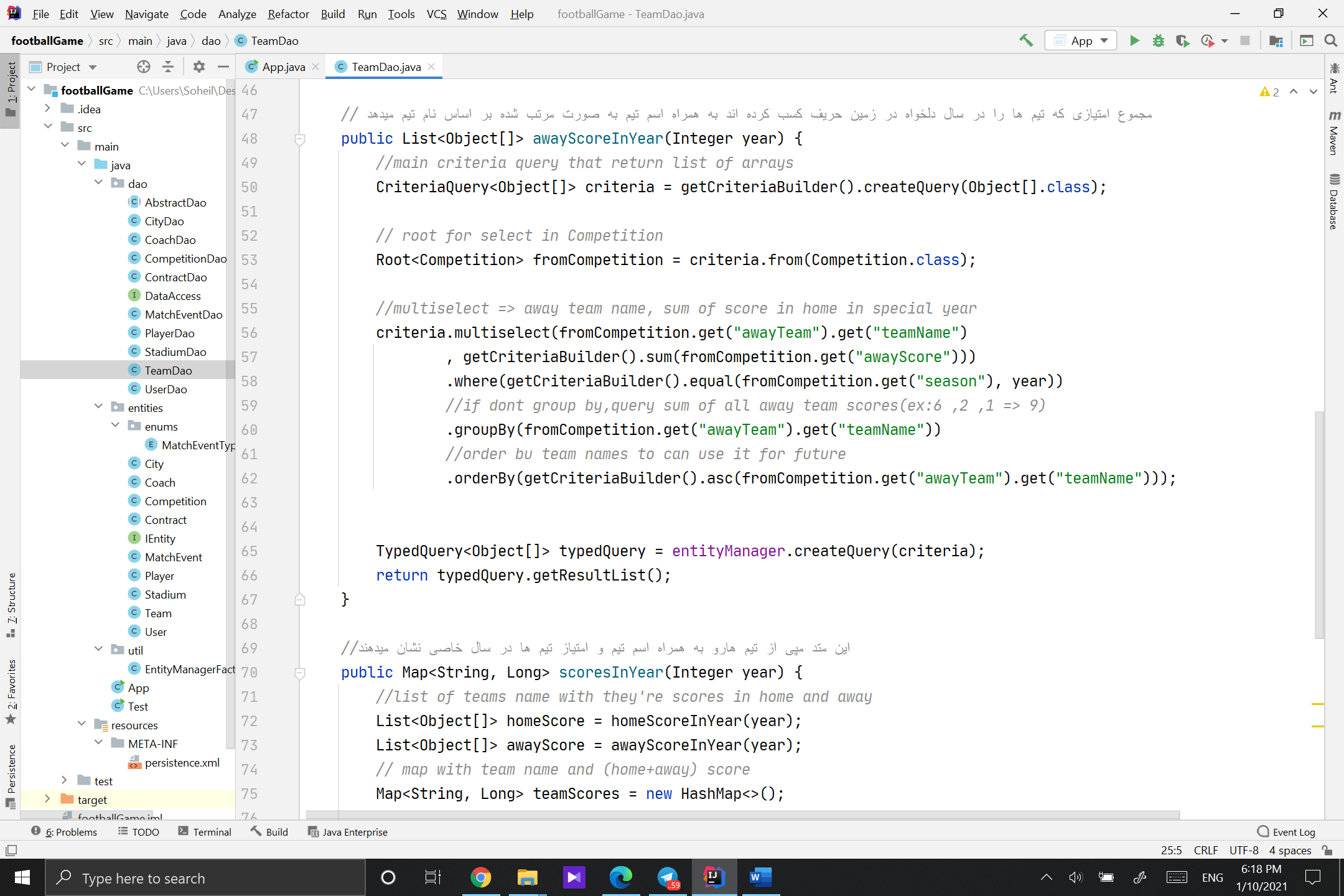


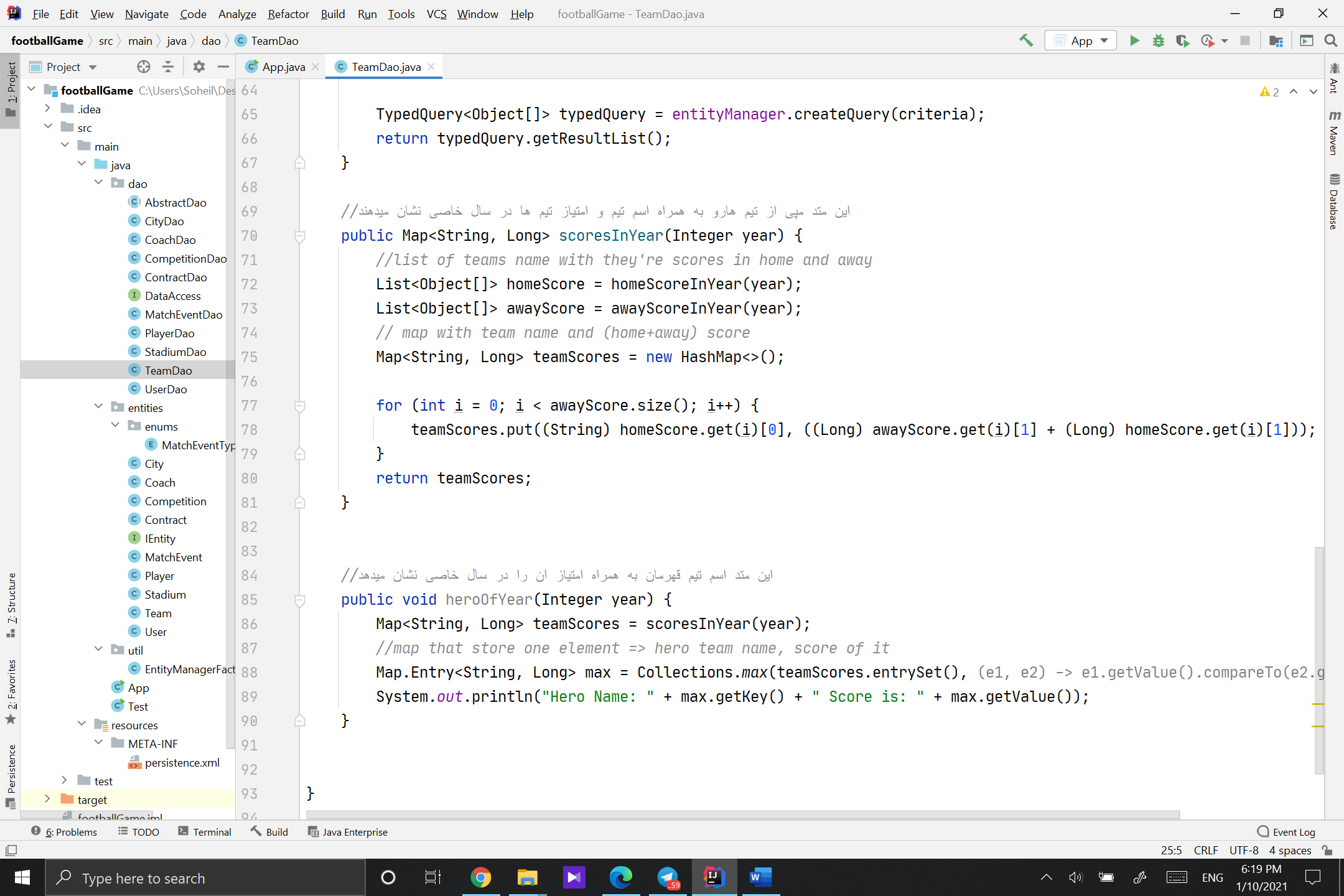
cityDao:



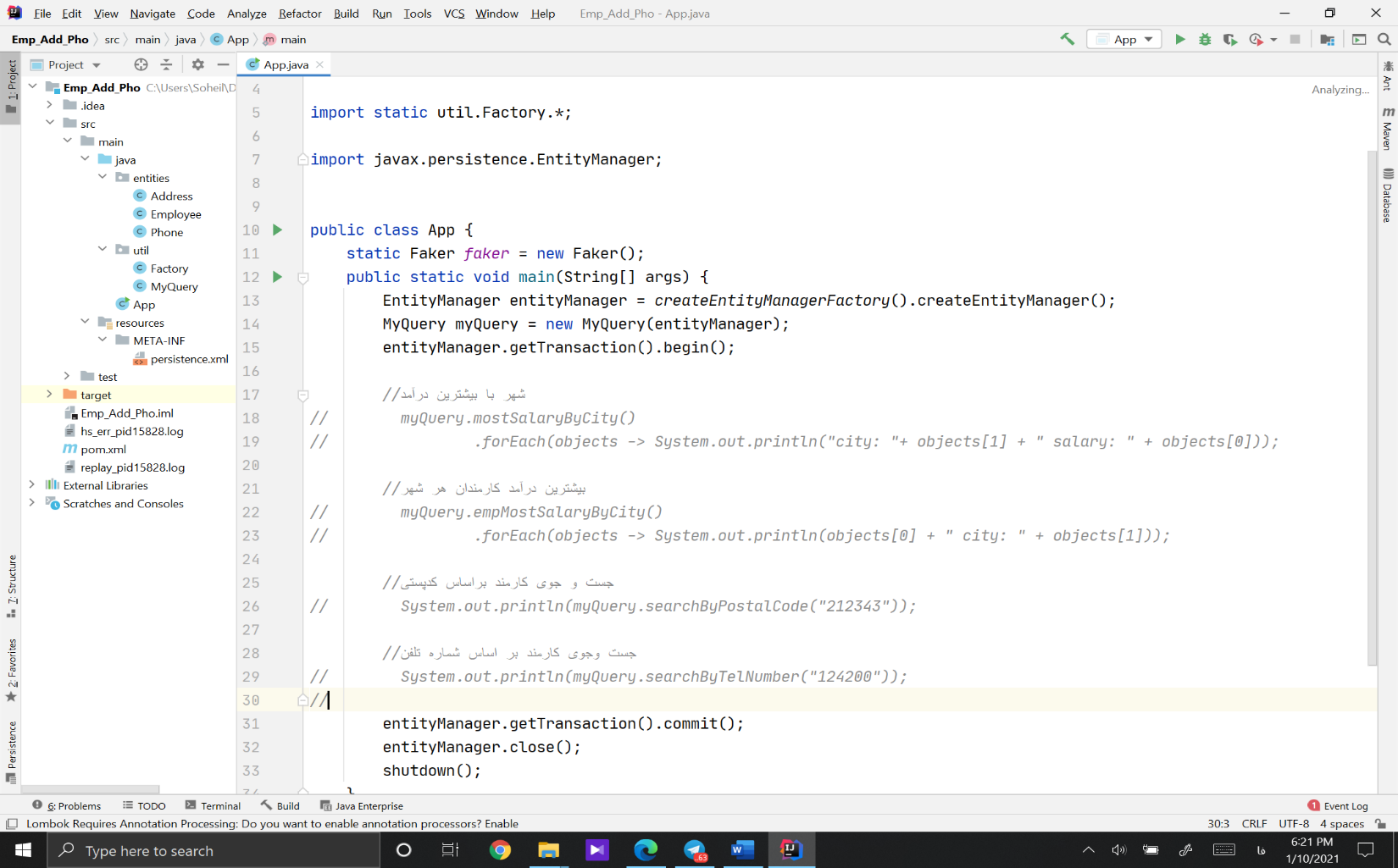
teamDao:

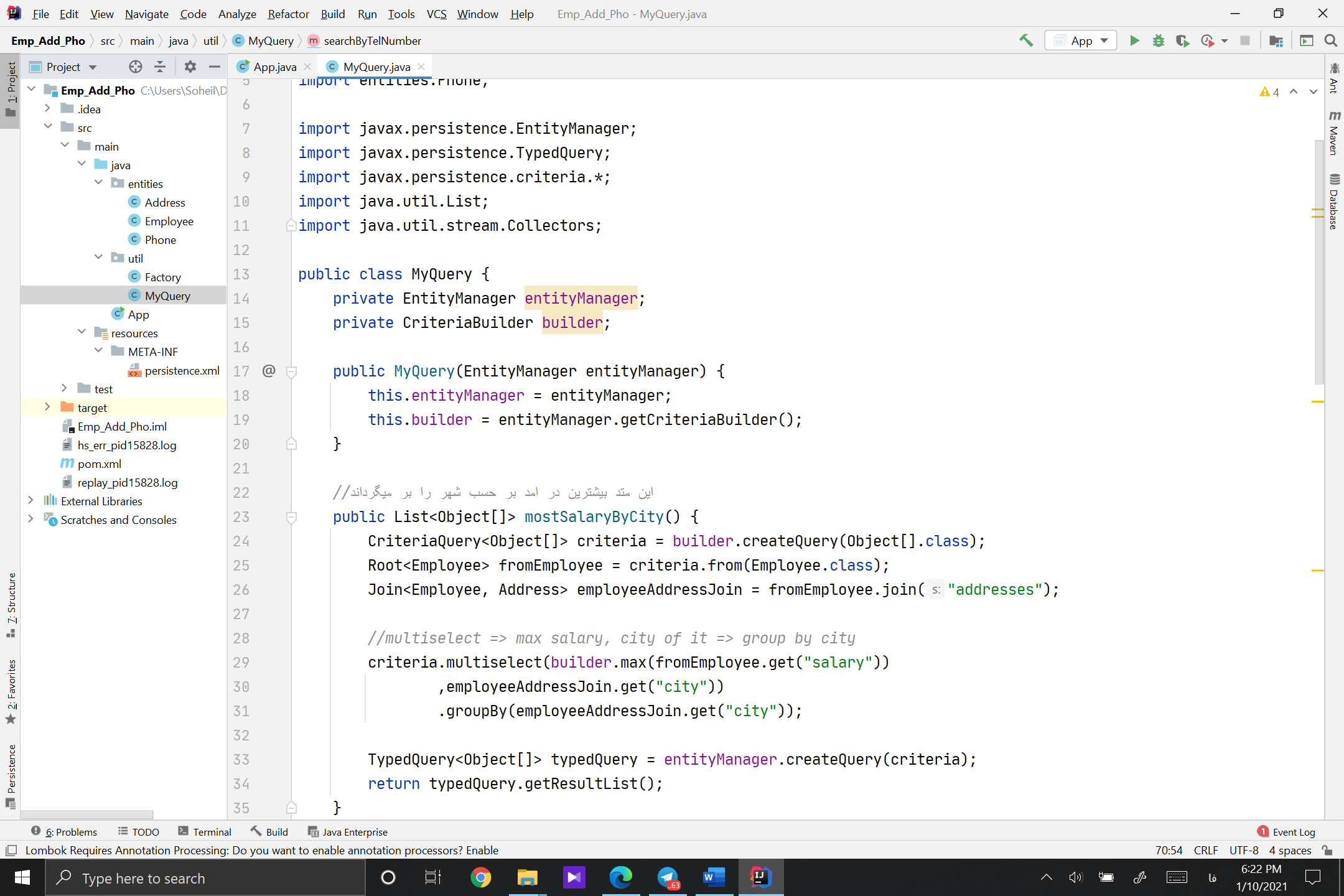


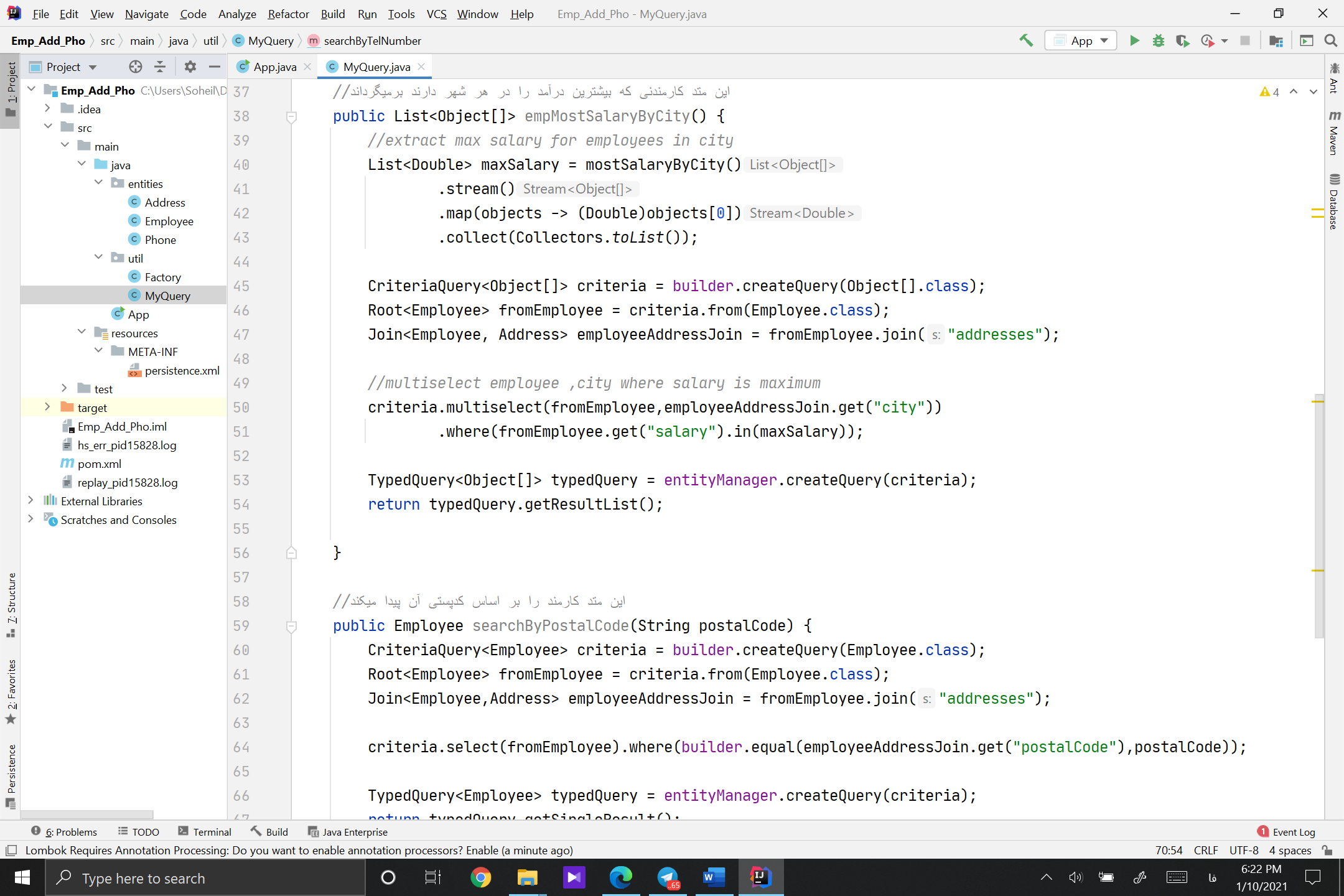


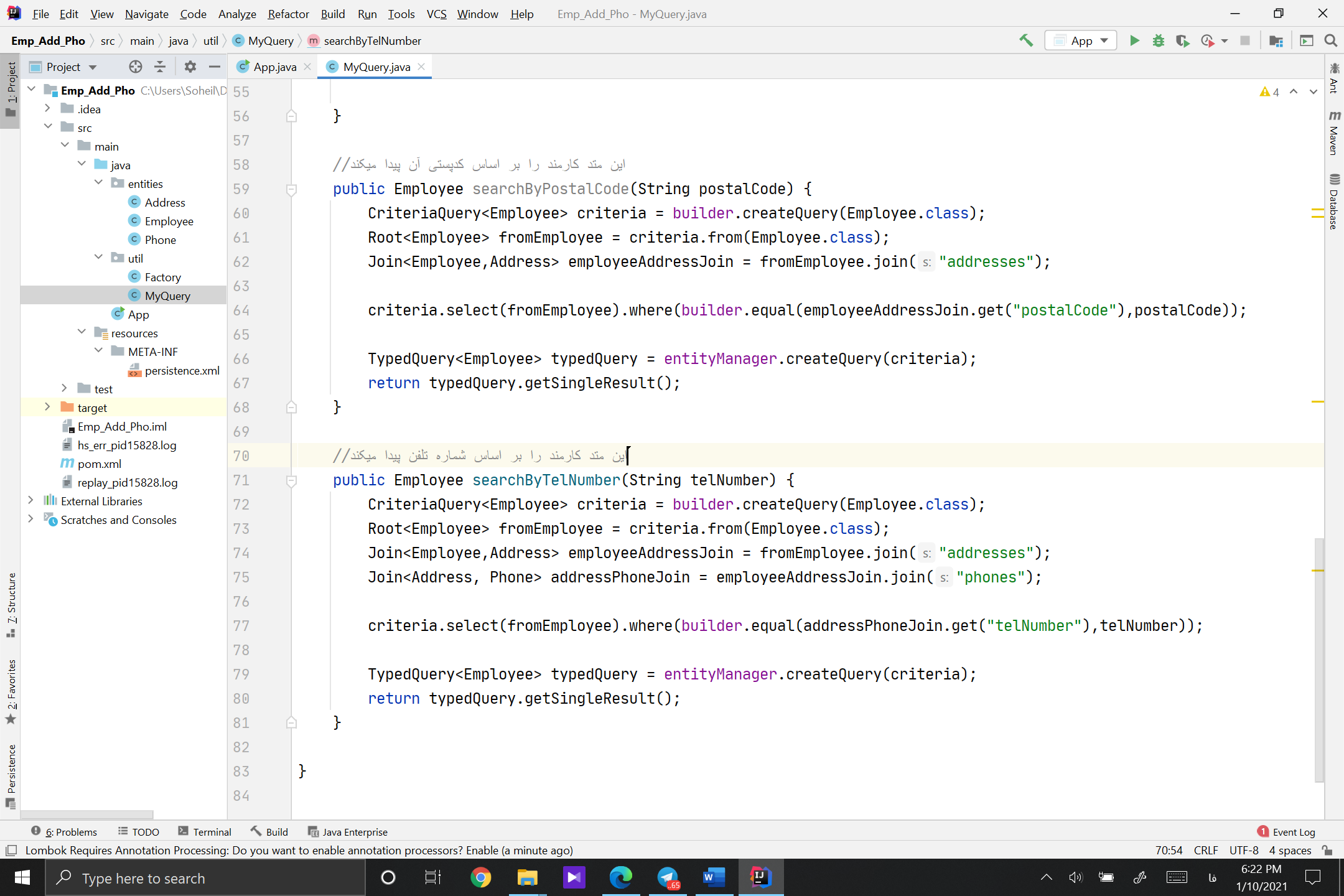


تمرین شماره 2:



****

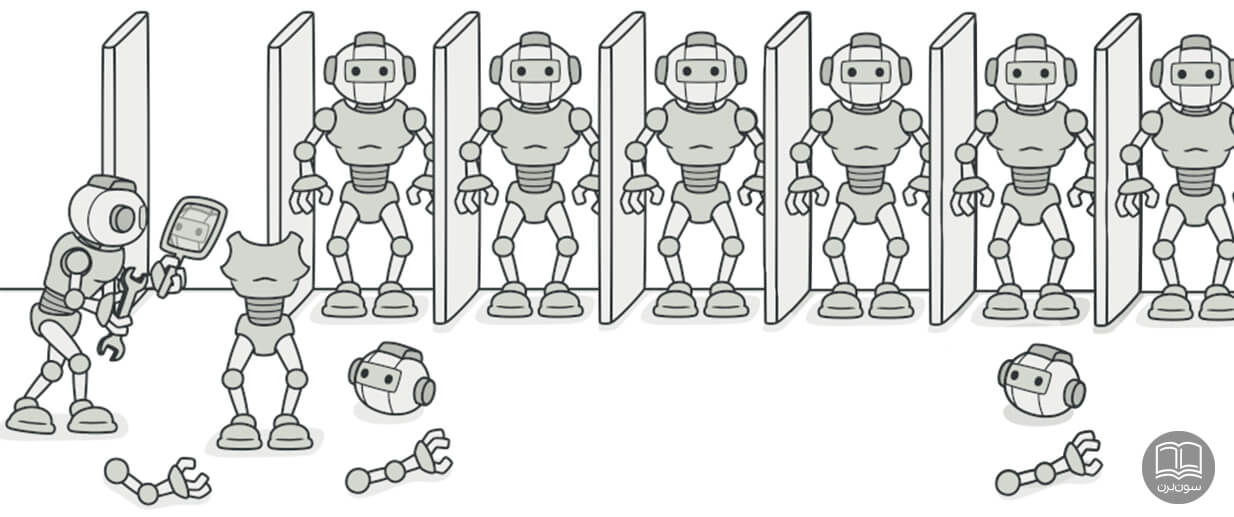
****

****

تمرین شماره 3:

الگوی پروتوتایپ(Prototype)

دیزاین پترن ها در سال 1994 توسط گروهی به نام Gang of Four در سه دسته عمومی طبقه بندی شدند. به آن دسته از الگوهای طراحی که با هدف مدیریت ایجاد اشیا توسعه یافته اند، الگوهای طراحی سازنده یا Creational می‌گویند. الگوی طراحی پروتوتایپ یا Prototype یکی از الگوی‌های طراحی سازنده است که به منظور جلوگیری از ساخت یک شی جدید استفاده می‌شود.

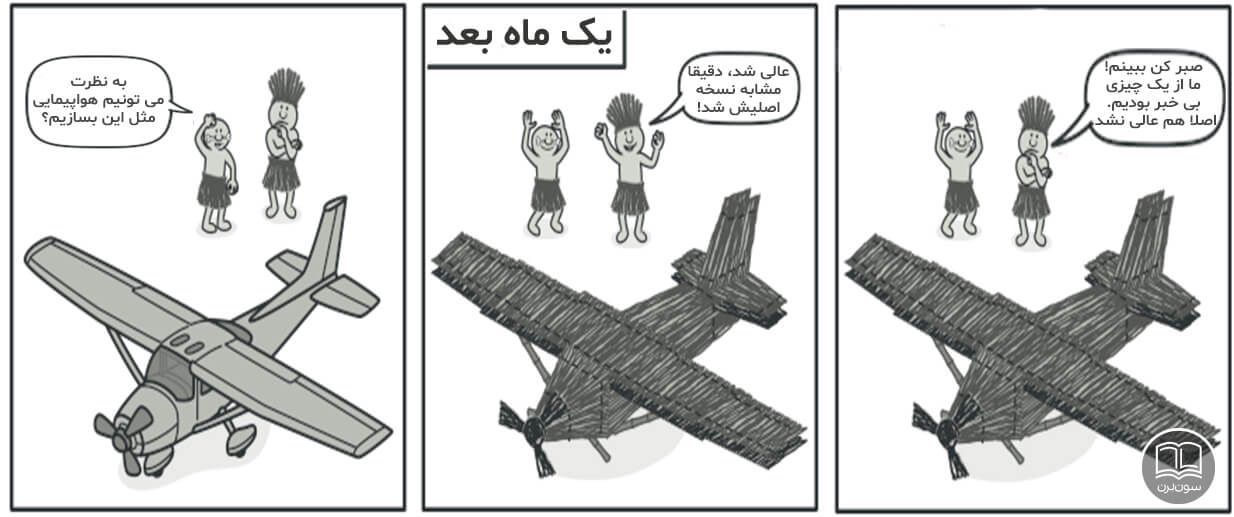


این الگوی طراحی راهکاری برای ایجاد یک شی جدید به واسطه کپی کردن آن از اشیا موجود دیگر ارائه می‌دهد. با استفاده از این تکنیک بدون اینکه وابستگی میان شی جدید و کلاسی که از آن ساخته شده است ایجاد شود، می‌توان اشیا جدیدی در نرم افزار تولید کرد. همچنین در مواردی که مواقع نیاز به ایجاد تغییرات در اشیا جدید وجود دارد تا بتوانیم اشیایی از یک نوع و با پارامترهایی متفاوت ایجاد کنیم.

مثلا تصور کنید بخواهید از یک شی ماشین چندین کپی ایجاد کنید با این تفاوت که هر کدام رنگی متفاوت داشته باشد. با استفاده از الگوی طراحی پروتوتایپ دیگر نیاز نیست که هر بار یک شی جدید از کلاس ایجاد کنید و سپس پارامترهای آن را مشخص کنید. زیرا شما می‌توانید از کلاس ماشین یک شی ایجاد کنید و سپس از آن به تعدادی که نیاز دارید کپی (Clone) تولید کنید.

**چرا باید از الگوی طراحی Prototype استفاده کنیم**

تصور کنید که تصمیم گرفته اید یک هواپیما بر اساس نمونه خارجی آن بسازید. برای بومی ساز این هواپیما تنها راهی که پیش روی شما خواهد بود این است که دقیقا از پارامترها و ویژگی‌های ساختاری نمونه خارجی آن آگاه باشید. اما از آنجایی که تکنولوژی هایی مربوط به ساخت هواپیماها انحصاری هستند و به راحتی در اختیار دیگر شرکت‌ها قرار نمی‌گیرند، این امر برای شما امکان پذیر نخواهد بود. اما تصور کنید که یک ماشین کپی سه بعدی غول پیکر برای شبیه سازی اشیا داشته باشید. با این کار می‌توانید هواپیما خارجی به آن دستگاه وارد کرده و هواپیما شبیه سازی خود را دریافت کنید. این ماشین شبیه ساز کاری شبیه الگوی طراحی پروتوتایپ انجام می‌دهد.



بنابراین اگر بخواهید از شی ای موجود در نرم افزار دقیقا یک کپی ایجاد کنید. ابتدا باید یک شی جدید از کلاس مورد نظر ایجاد کنید. سپس باید تمام پارامترها و ویژگی‌های شی اصلی را دریافت کرده و بر روی جسم جدید اعمال کنید. اما این روش همیشه به درستی عمل نخواهد کرد زیرا بعضی از پارامترها و ویژگی‌های یک شی ممکن است به صورت خصوصی یا Private تعریف شده باشند. در این صورت دیگر امکان دسترسی به آن‌ها از خارج از کلاس امکان پذیر نخواهد بود.

بنابراین یکی از مشکلات کپی کردن اشیا در برنامه نویسی به صورت مستقیم، عدم دسترسی به برخی از پارامترهای خصوصی است. پس کپی کردن یک شی خارج از کلاسش همیشه امکان پذیر نخواهد بود. از طرفی دیگر به دلیل اینکه شما اشیای کپی شده را بر پایه کلاس‌های شی مبدا ایجاد می‌کنید، اشیای جدید به این کلاس‌ها وابسته می‌شوند و این کار باعث ایجاد وابستگی‌های زیاد در بین کلاس‌ها خواهد شد.

**راه حل الگوی طراحی پروتوتایپ**

در الگوی طراحی نمونه اولیه یک Interface مشترک برای تمام اشیایی که باید کپی شوند در نظر گرفته می‌شود. این Interface به شما اجازه می‌دهد که یک شی را بدون نیاز به دوباره نویسی کدها در کلاس شی، کپی کنید. معمولا، چنین Interface شامل تنها یک تابع به اسم Clone (کپی) هستند. پس با پیروی کلاس‌های مربوط به اشیا از این Interface دیگر نیازی به نوشتن توابع اضافه و یا استفاده از ایجاد مجدد یک شی نخواهیم داشت. برای این کار فقط کافی است تابع Clone را در کلاس‌ها پیاده سازی کنیم.

پیاده سازی تابع Clone در همه کلاس‌ها بسیار شبیه به هم است. این تابع وظیفه دارد تا یک شی از کلاس فعلی ایجاد کند و سپس تمام مقادیر پارامترهای آن را به شی جدید منتقل می‌کند. همچنین مزیت دیگری که این الگوی طراحی در اختیار شما می‌گذارد این است که حتی می‌توانید پارامترهای Private را کپی کنید. زیرا اکثر زبان‌های برنامه نویسی اجازه می‌دهند که اشیاء به فیلدهای خصوصی از اشیاء دیگر که متعلق به یک کلاس هستند دسترسی پیدا کنند.

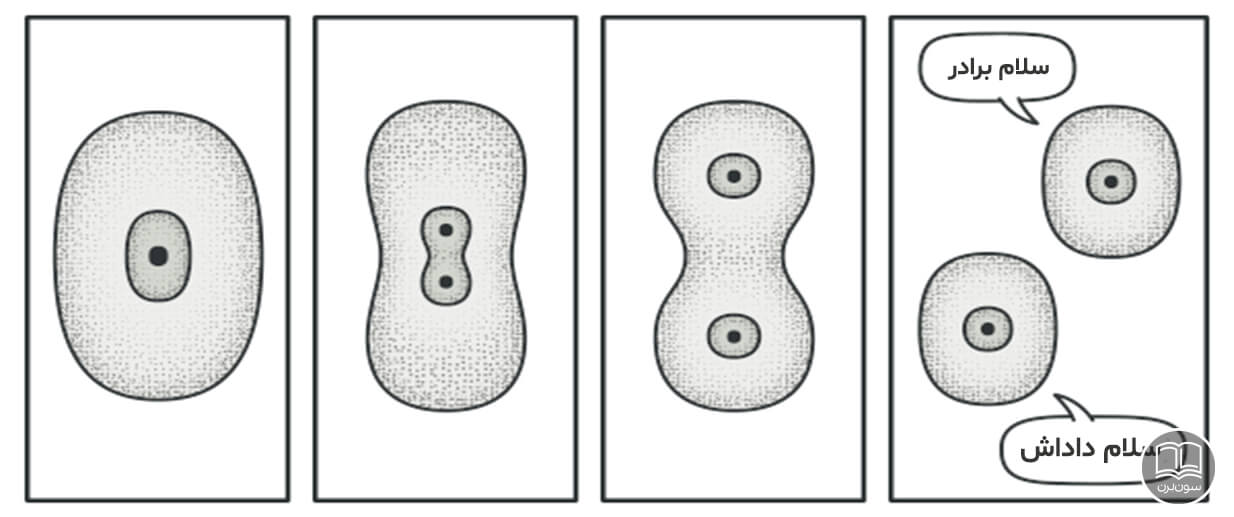
به اشیایی که کلاس‌های آن‌ها از قابلیت کپی شدن پشتیبانی کنند یک نمونه اولیه یا Prototype نامیده می‌شوند. همچنین هنگامی که اشیا دارای ده‌ها پارامتر و صدها تنظیمات باشند، می‌توانید از اشیا کپی شده به عنوان جایگزینی برای کلاس‌های فرزند استفاده شوند. با این کار در نرم افزار دیگر نیازی به ساخت کلاس‌های فرزند از کلاس اصلی نخواهید داشت. زیرا می‌توانید اشیا ساخته شده بر اساس کلاس اصلی را متناسب با نیازهایی که دارید تغییر دهید.



بنابراین اگر بخواهیم روش کار این الگو را جمع بندی کنیم باید گفت که شما در ابتدا باید مجموعه ای از اشیا را ایجاد کنید که به شیوه‌های مختلف پیکربندی شده اند. سپس در هر زمانی که به یک شی با یکی از پیکربندی‌های تنظیم شده نیاز داشته باشید، کافی است یک نمونه اولیه را بر اساس آن ایجاد کنید و دیگر نیازی به ساخت یک شی جدید از ابتدا نخواهید داشت.

**مثالی از الگوی طراحی پروتوتایپ در دنیای واقعی**

در دنیای واقعی، نمونه‌های اولیه برای انجام آزمایش‌های مختلف قبل از شروع تولید انبوه محصولات استفاده می‌شوند. اما در این الگوی طراحی، نمونه‌های اولیه به منظور انجام آزمایشات مختلف یا کنترل کیفیت استفاده نمی‌شوند. نمونه‌های ساخته شده در این تکنیک به صورت مستقیم در بخش‌های مورد نیاز در نرم افزار استفاده می‌شوند. بنابراین، تقریبا بهترین مثال از این الگوی طراحی در دنیای واقعی فرآیند تقسیم سلولی است.



تصور کنید که یک کلاس به اسم سلول (Cell) در نرم افزار تعریف شود. برای اینکه این سلول به یک موجود تکامل یافته تبدیل شود، نیاز به تکثیر و نمونه سازی دارد. بنابراین به هزاران یا میلیون‌ها شی از این کلاس نیاز خواهیم داشت تا اندام‌های مختلف یک موجود زنده را تشکیل دهند و هر کدام به وظیفه خاص خود بپردازد. حال تصور کنید، اگر می‌خواستید همه این اشیا را به صورت مستقیم و یک به یک بسازید و پارامترهای آن‌ها را مقدار دهی کنید، چه میزان از زمان شما مصرف می‌شد.

بر این اساس بهترین راه برای انجام این کار، به کارگیری الگوی طراحی پروتوتایپ خواهد بود. زیرا در این الگو سلول‌ها بر اساس سلول اولیه کپی خواهند شد و دیگر نیازی به ساخته شدن سلول‌ها و مقدار دهی پارامترهای آن‌ها بر اساس سلول اولیه به صورت یک به یک نخواهد بود. با توجه به این نکته که در فرآیند تقسیم سلولی، سلول‌های جدید از سلول اولیه ایجاد می‌شوند می‌توان اینگونه گفت که سلول اصلی به عنوان یک نمونه اولیه یا Prototype عمل می‌کند.

**کاربردهای الگوی طراحی Prototype**



به کارگیری الگوی طراحی پروتوتایپ در طراحی نرم افزارها، باعث کاهش میزان کدنویسی خواهد شد. مزیت اصلی این دیزاین پترن افزایش سرعت کپی کردن یک شی از شی‌های است. زیرا این عمل خیلی سریع‌تر از ساخت اشیا می‌باشد. در عملیات کپی کردن اشیا تابع سازنده آن‌ها (Constructor) دیگر اجرا نخواهد شد. الگوی طراحی سازنده در موارد زیر کاربرد خواهد داشت:

1. برای کاهش تعداد کلاس‌های فرزند (Subclass) استفاده می‌شود. همچنین با این روش می‌توان اشیا ساخته شده را متناسب با نیازهای مختلف پیکربندی کرد.
2. به منظور جلوگیری از ایجاد وابستگی میان شی کپی شده به کلاس مورد نظر استفاده می‌شود.
3. زمانی که بخواهیم از یک کلاس با پارامترهای Private زیاد شی ایجاد کنیم.