



به نام خدا

دانشگاه اصفهان
دانشکده مهندسی کامپیوتر

تکلیف اول درس برنامه نویسی دستگاه های سیار

استاد درس:

دکتر شیرمحمدی

نام و نام خانوداگی:

محمدجواد صالحی

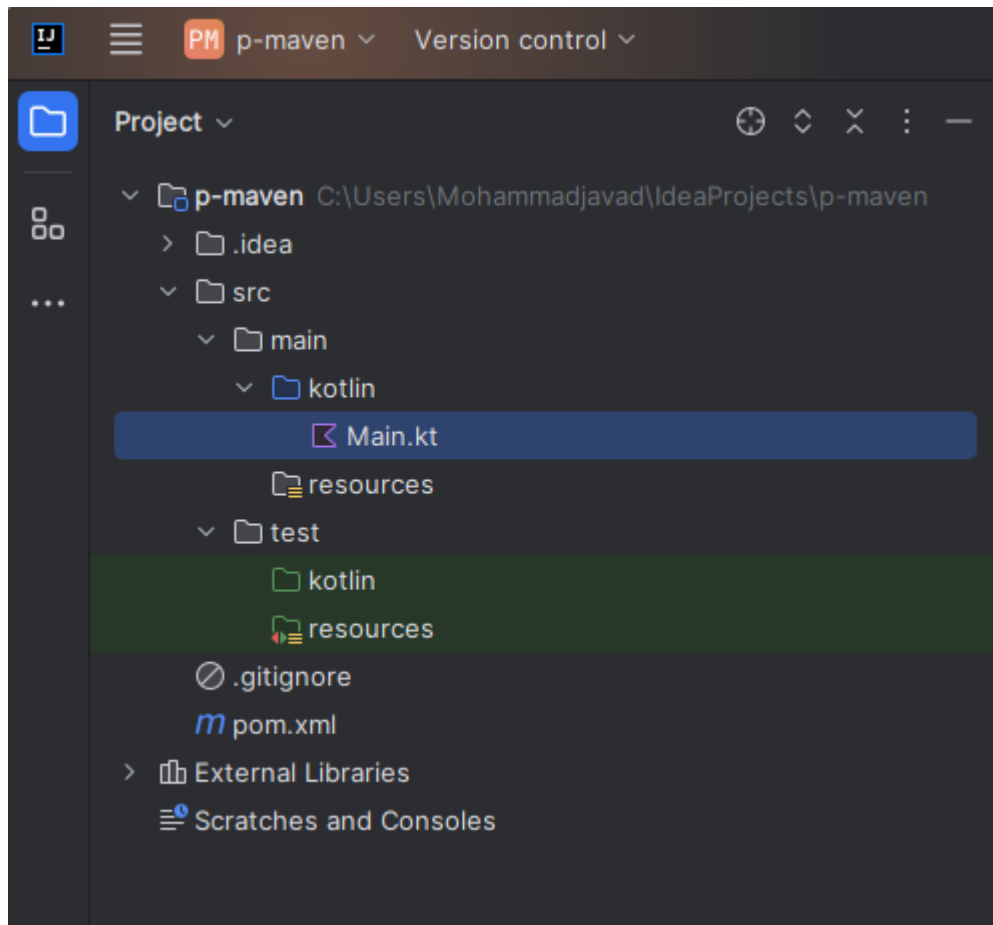
شماره دانشجویی:

۹۹۳۶۲۳۰۲۷

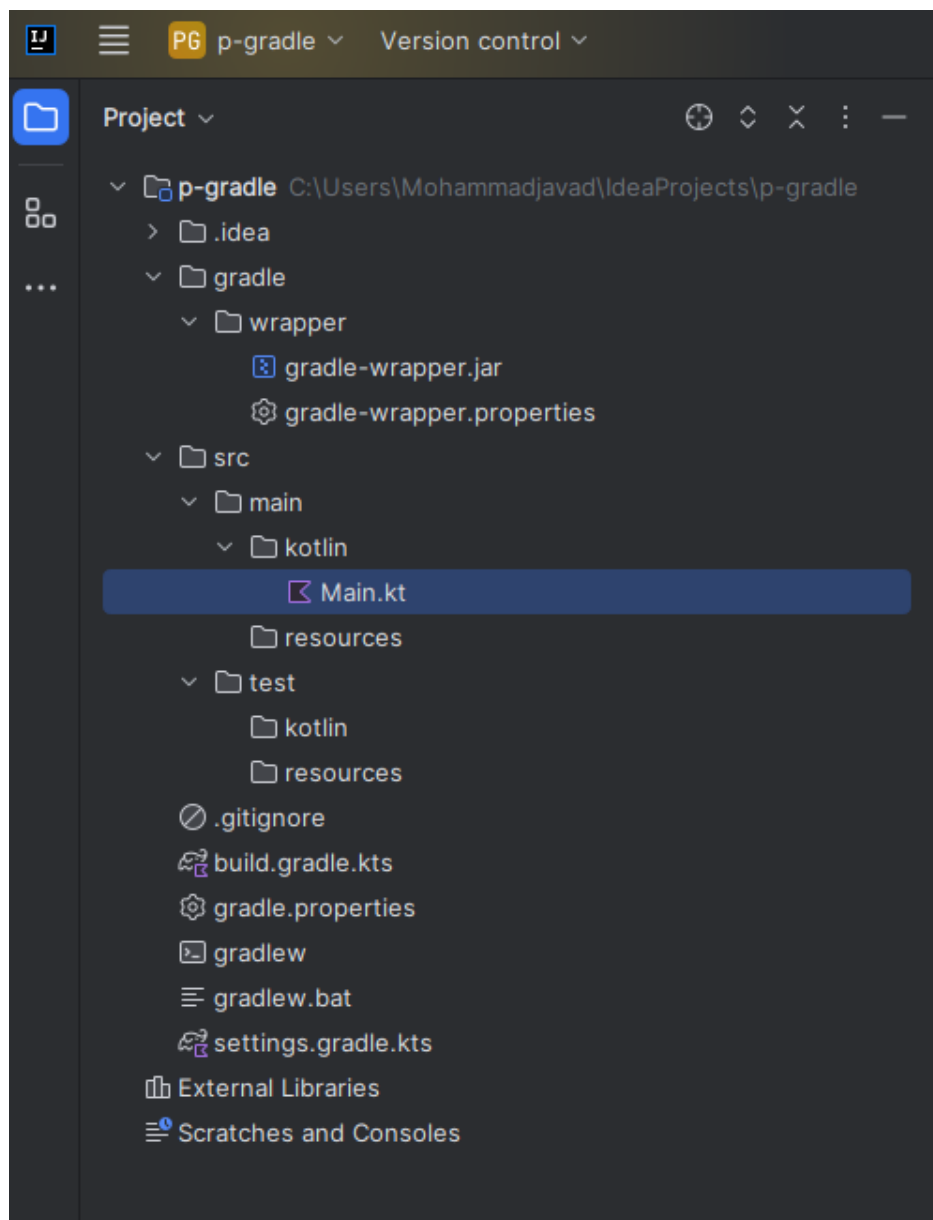
زمستان ۱۴۰۲

- بررسی ساختار فایل های پروژه

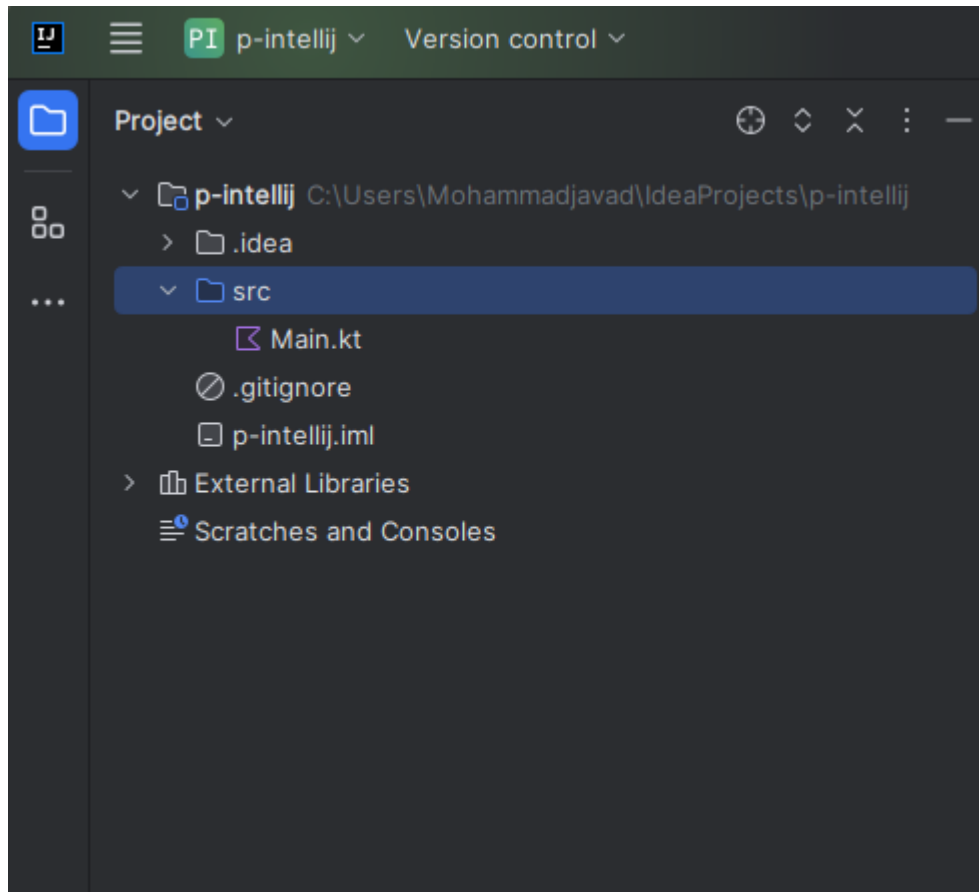
ساختار فایل های پروژه با بیلد سیستم Maven



ساختار فایل های پروژه با بیلد سیستم Gradle



ساختار فایل های پروژه با بیلد سیستم IntelliJ



- در IntelliJ IDEA، پروژه‌ها معمولاً از قاعده های طرح‌بندی (IntelliJ layout conventions) با یک دایرکتوری `idea` حاوی تنظیمات خاص IDE، در کنار دایرکتوری‌های `src` برای کد اصلی و تست پیروی می‌کنند. همچنین با اجرای پروژه فولدر `out` که حاوی فایل های اجرایی ست ایجاد می شود.
- پروژه‌های Gradle از قاعده های Gradle پیروی می‌کنند که شامل یک فایل `build.gradle` یا `build.gradle.kts` برای پیکربندی‌ها و دایرکتوری‌های `src/main/kotlin` و `src/test/kotlin` به ترتیب برای کد منبع و تست است. در این بیلد سیستم با اجرای پروژه فولدر `build` که حاوی فایل های اجرایی ست ایجاد می شود.
- پروژه‌های Maven دارای طرح‌بندی استاندارد شده با فایل `pom.xml` هستند که پیکربندی‌های پروژه را تعریف می‌کند (در ادامه توضیح داده شده است) و کد منبع در `src/main/kotlin` و `src/test/kotlin` قرار دارد. هر سیستم دایرکتوری های خاصی را برای ساخت وابستگی ها ایجاد می کند. در این بیلد سیستم با اجرای پروژه فولدر `target` که حاوی فایل های اجرایی ست ایجاد می شود.

- فایل pom یا Project Object Model واحد اساسی کار در Maven است. این فایل در واقع یک فایل XML است که حاوی اطلاعاتی درباره پروژه و جزئیات پیکربندی است که توسط Maven برای ساخت پروژه استفاده شده است. این اطلاعات شامل وابستگی ها، دایرکتوری ساخت، دایرکتوری منبع، دایرکتوری منبع تست، افزونه، اهداف، نسخه پروژه، توضیحات و... است. هنگام اجرای یک کار یا هدف، Maven به دنبال POM می گردد. POM را می خواند، اطلاعات پیکربندی مورد نیاز را دریافت می کند، سپس هدف را اجرا می کند.

Element	Description
project	It is the root element of pom.xml file.
modelVersion	It is the sub element of project. It specifies the modelVersion. It should be set to 4.0.0.
groupId	It is the sub element of project. It specifies the id for the project group.
artifactId	It is the sub element of project. It specifies the id for the artifact (project). An artifact is something that is either produced or used by a project. Examples of artifacts produced by Maven for a project include: JARs, source and binary distributions, and WARs.
version	It is the sub element of project. It specifies the version of the artifact under given group.

- **<project>**: عنصر ریشه فایل POM. تمام عناصر دیگر در این عنصر قرار دارند.
- **<modelVersion>**: نسخه ای از مدل POM را که فایل به آن پایبند است را مشخص می کند. برای پروژه های مدرن Maven، این معمولاً روی 4.0.0 تنظیم می شود.
- **<groupId>**: گروه یا سازمانی را که پروژه به آن تعلق دارد را مشخص می کند. این اغلب بر اساس نام بسته پروژه است.
- **<artifactId>**: نام پروژه در حال ساخت را تعریف می کند.
- **<version>**: نسخه ای از پروژه در حال ساخت را مشخص می کند.
- **<packaging>**: نوع بسته بندی پروژه را نشان می دهد (به عنوان مثال، jar، war، pom).
- **<dependencies>**: وابستگی های مورد نیاز پروژه را فهرست می کند. هر عنصر **<dependency>** یک وابستگی را مشخص می کند که پروژه به آن متکی است، از جمله **groupId**، **artifactId** و نسخه آن.
- **<build>**: حاوی جزئیات پیکربندی برای ساخت پروژه است. این شامل عناصری مانند **<plugins>** برای تعیین افزونه های Maven برای استفاده در طول فرآیند build است.
- **<properties>**: به تعریف ویژگی هایی اجازه می دهد که می توانند در جای دیگری در فایل POM ارجاع شوند. این برای اجتناب از افزونگی و برای مدیریت مقادیر پیکربندی به صورت متمرکز مفید است.
- **<repositories>** و **<pluginRepositories>**: این عناصر مخازنی را تعریف می کنند که Maven می تواند به ترتیب وابستگی ها و افزونه ها را از آنها دانلود کند.
- **<profiles>**: به تعریف مجموعه های مختلفی از گزینه های پیکربندی اجازه می دهد که بر اساس شرایط مختلف (مانند متغیرهای محیط، سیستم عامل) فعال شوند.

منابع:

- 1- <https://maven.apache.org/guides/introduction/introduction-to-the-pom.html>
- 2- <https://chat.openai.com/>
- 3- <https://www.javatpoint.com/maven-pom-xml>