

# آزمایش سوم آزمایشگاه مهندسی نرم افزار

دانشگاه صنعتی شریف  
دانشکده مهندسی کامپیوتر

سینا الهی منش - ۹۸۱۰۱۱۶۹  
سارا زاهدی موحد - ۹۸۱۷۰۸۴۹

تابستان ۱۴۰۲

لینک ریپازیتوری گیت‌هاب: <https://github.com/sarazm2000/BDD>

## مقدمه

BDD یا Behavior Driven Development یک روش توسعه نرم‌افزاری است که تمرکز اصلی آن بر رفتار سیستم است. در این روش، تیم‌های توسعه نرم‌افزاری با همکاری با مشتریان و تیم‌های تضمین کیفیت، ابتدا سناریوها و توصیفاتی را از رفتارهای مختلف سیستم ایجاد می‌کنند.

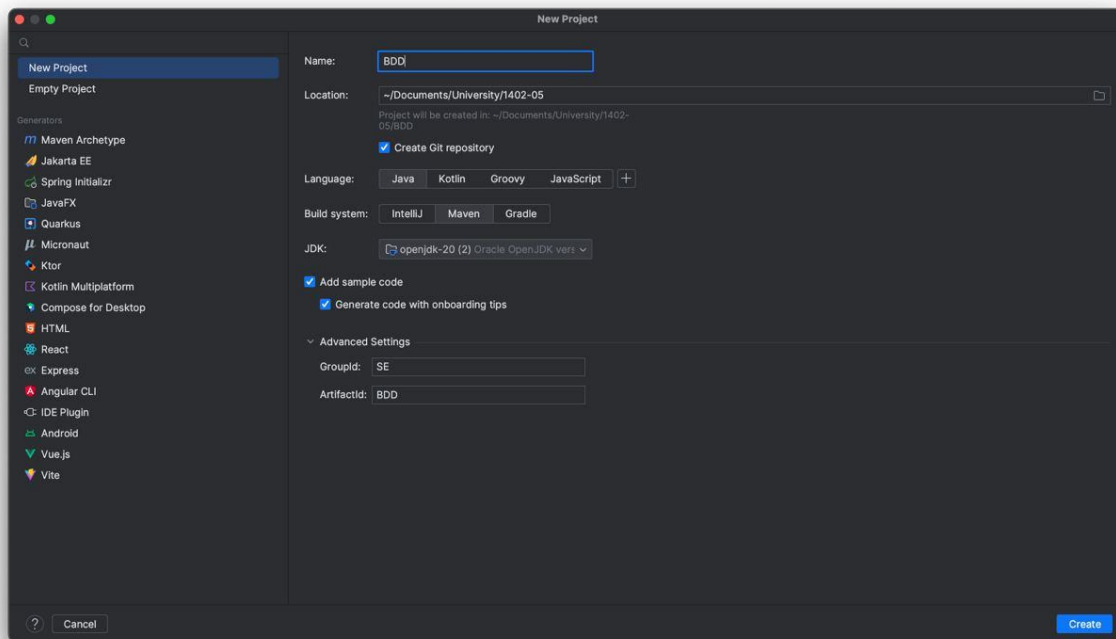
این سناریوها به زبان ساده‌تری نوشته می‌شوند و از مشتریان و تیم‌های تضمین کیفیت برای تأیید و تفسیر استفاده می‌شوند. سپس، تیم‌های توسعه نرم‌افزاری بر اساس این سناریوها کدها و تست‌های مرتبط را پیاده‌سازی می‌کنند.

BDD به عنوان یک روش همکاری، به توسعه‌دهندگان و مشتریان کمک می‌کند تا درک بهتری از نیازها و انتظارات دیگران داشته باشند و باعث ایجاد نرم‌افزارهای با کیفیت‌تر و هماهنگ‌تر می‌شود. این روش به عنوان یک ابزار موثر در ارتقاء همکاری و ارتباطات بین اعضای تیم توسعه می‌تواند بهبود محصول نهایی را تسهیل کند.

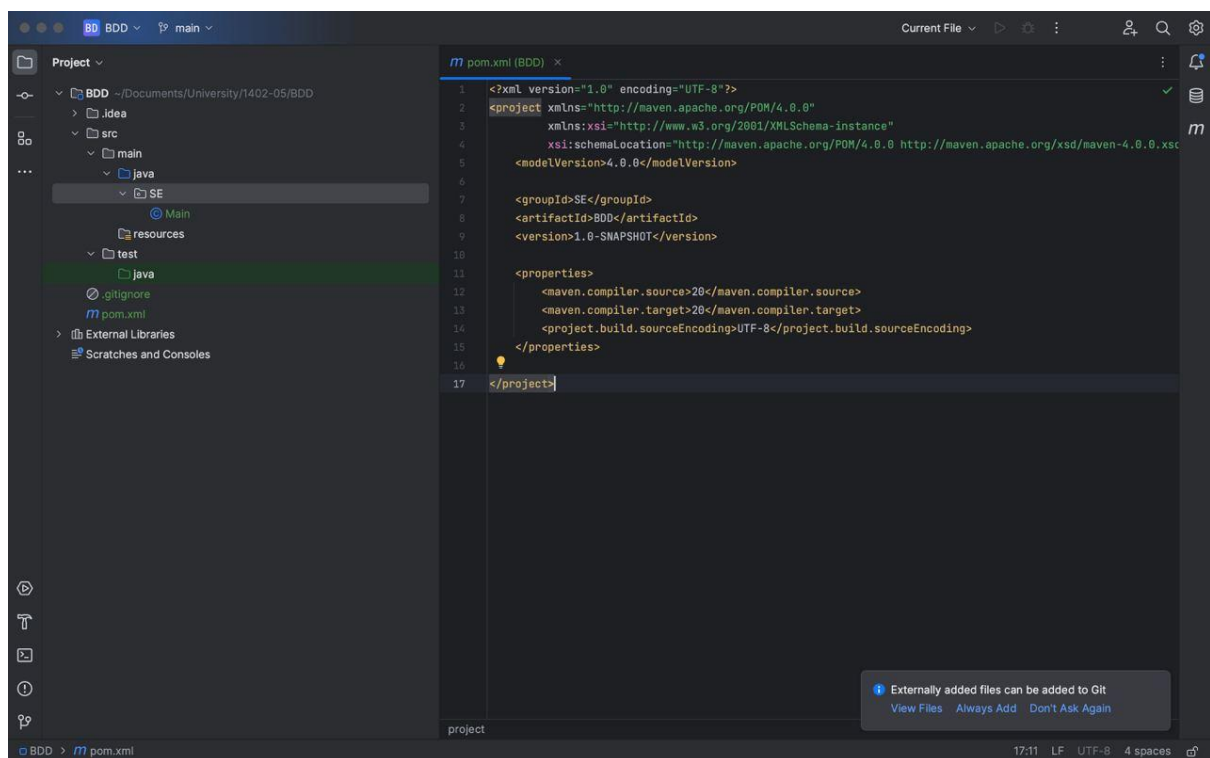
## شرح آزمایش

در این بخش آزمایش انجام شده را به صورت مرحله به مرحله همراه با تصویر توضیح می‌دهیم.

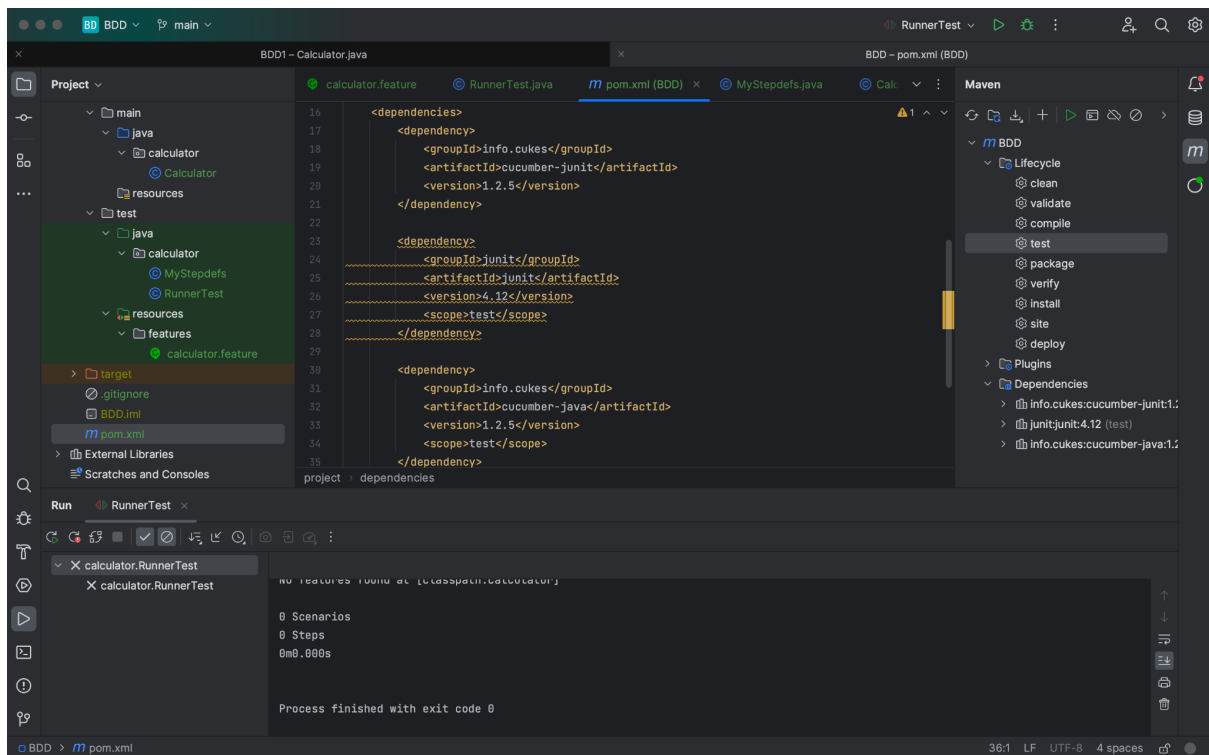
گام اول ایجاد پروژه maven



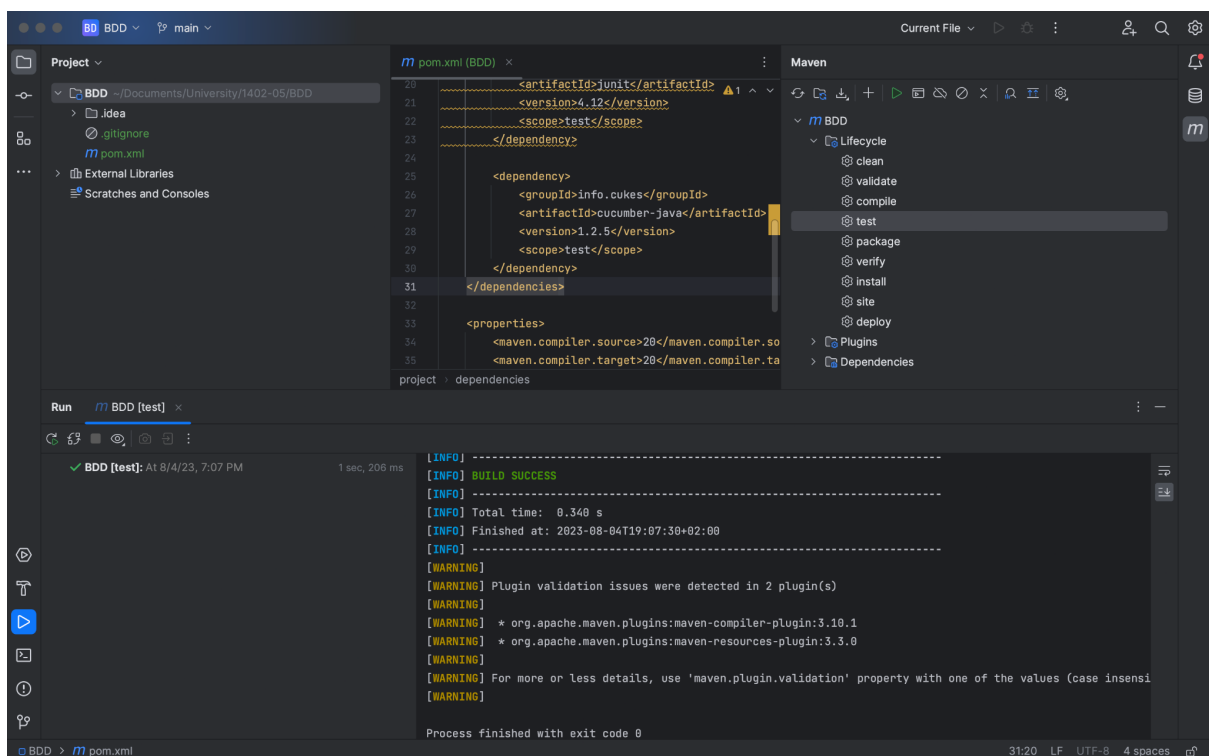
پس از ایجاد پروژه یک پروژه خالی برایمان ساخته میشود.



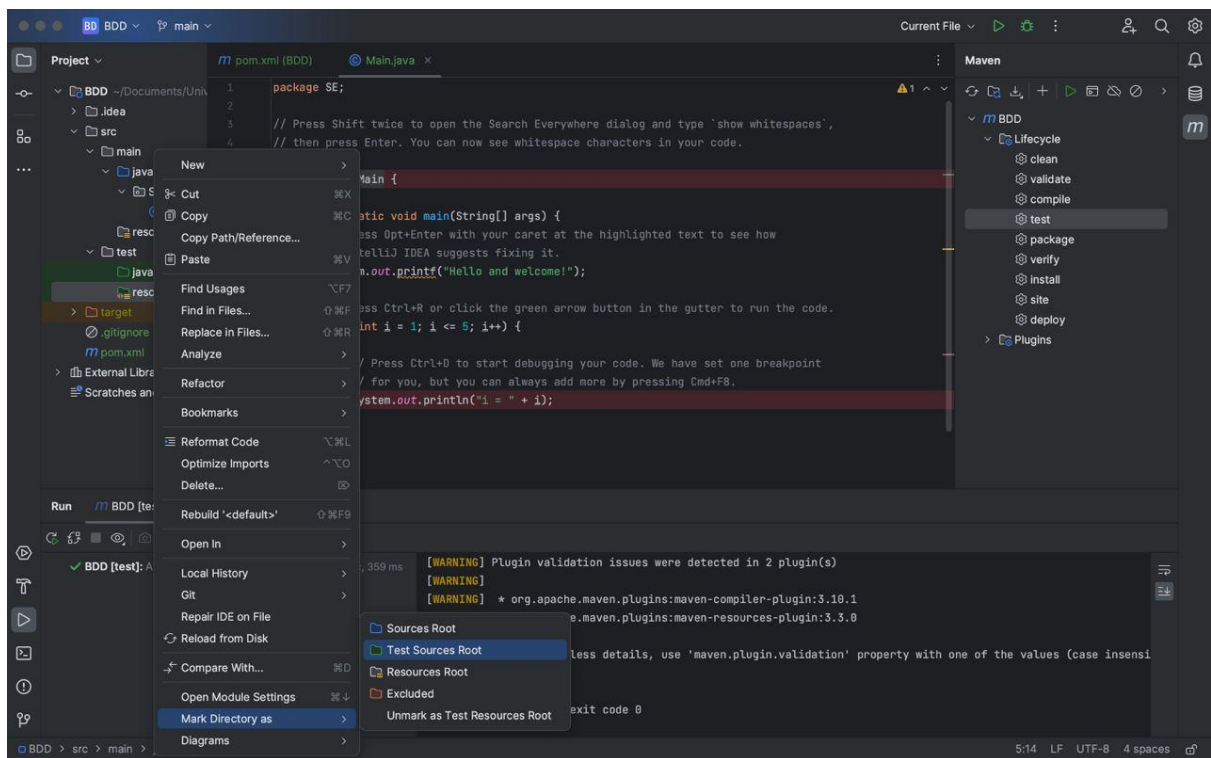
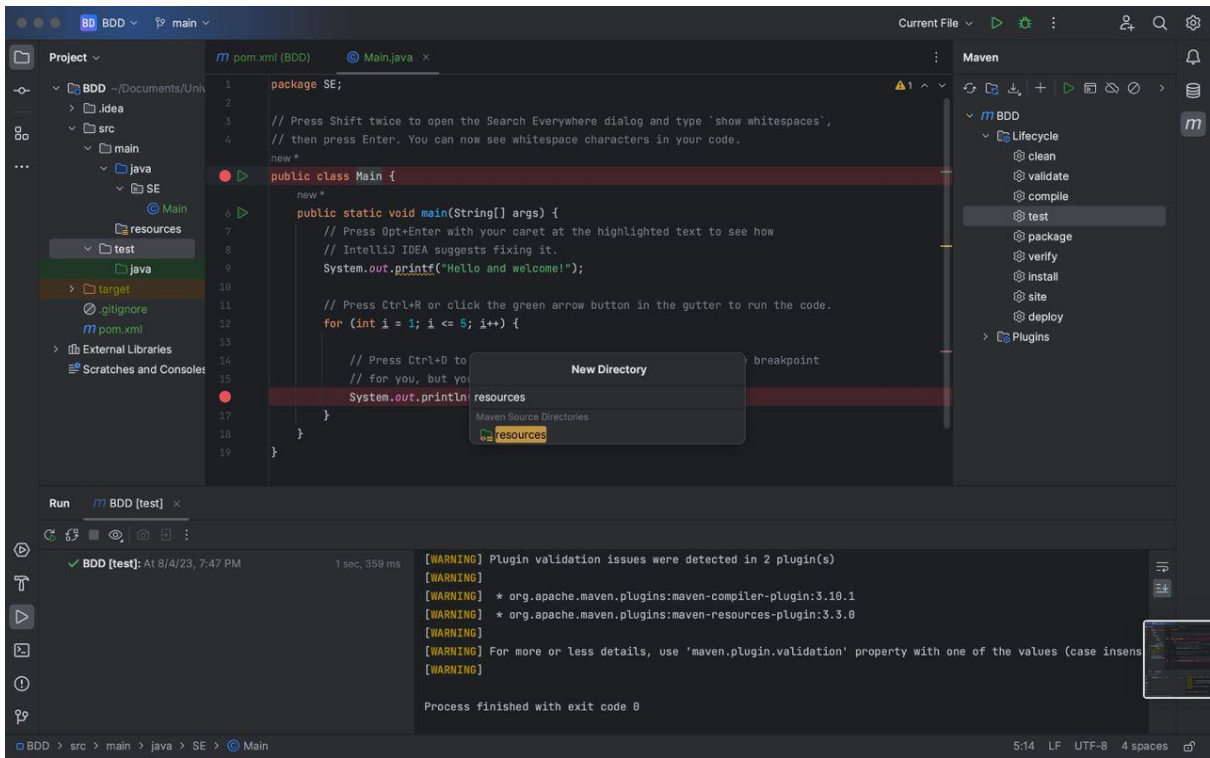
دیپندسی های لازم را به فایل pom.xml اضافه میکنیم.

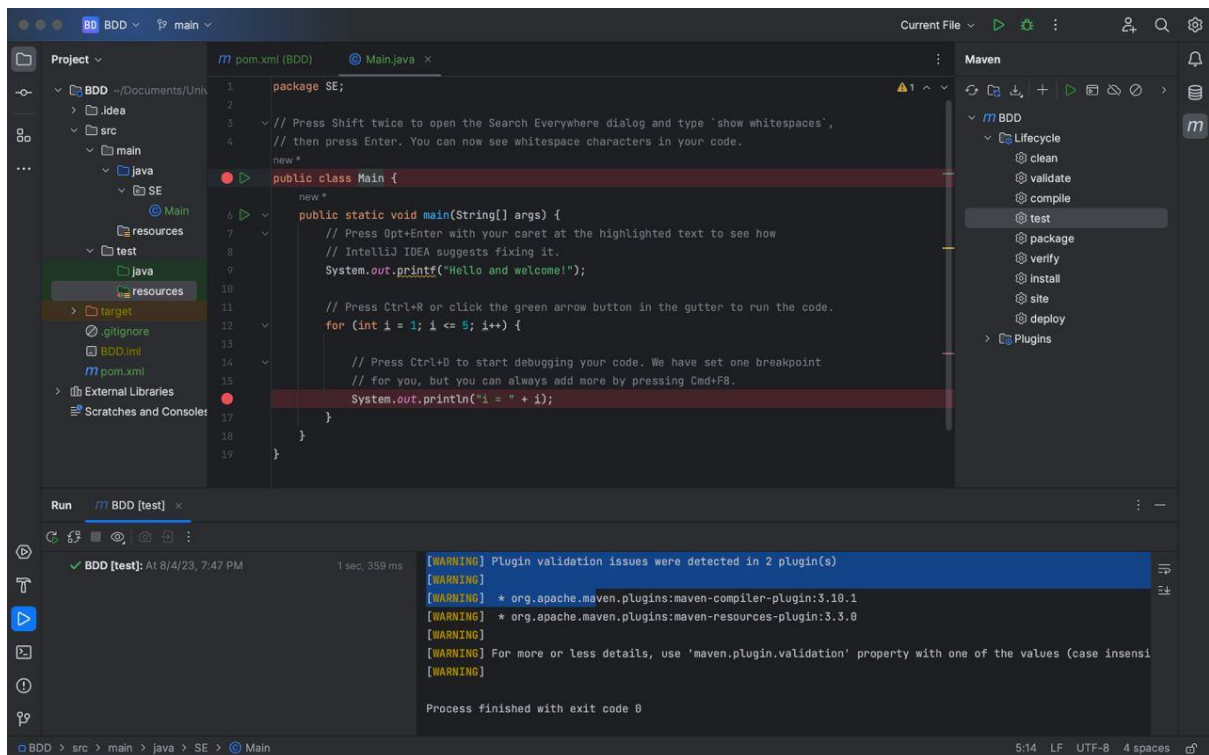


پس از افزودن دیپندنسی ها در قسمت maven و lifecycle، test را اجرا میکنیم و با پیام BUILD SUCCESS مواجه میشویم.

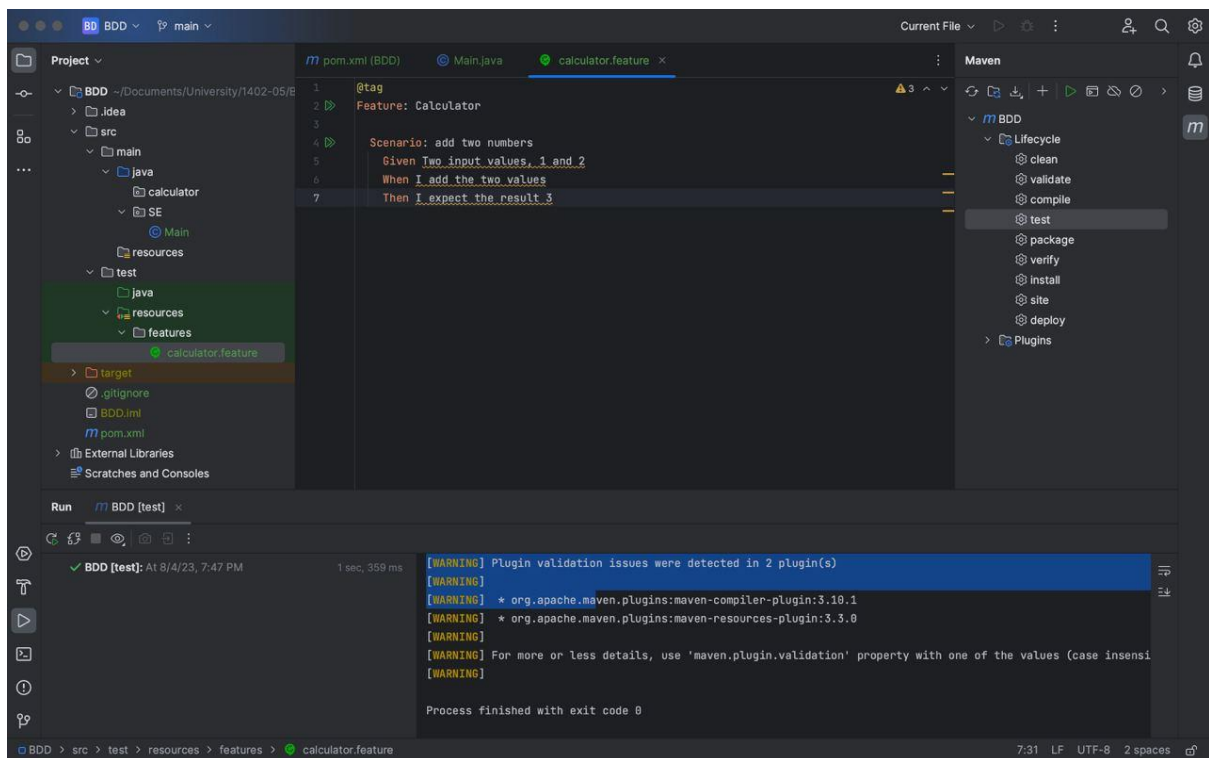


در مرحله بعدی یک پکیج در پکیج test با نام resources ایجاد میکنیم و آن را Make Directory as Test Resources Root میکنیم.

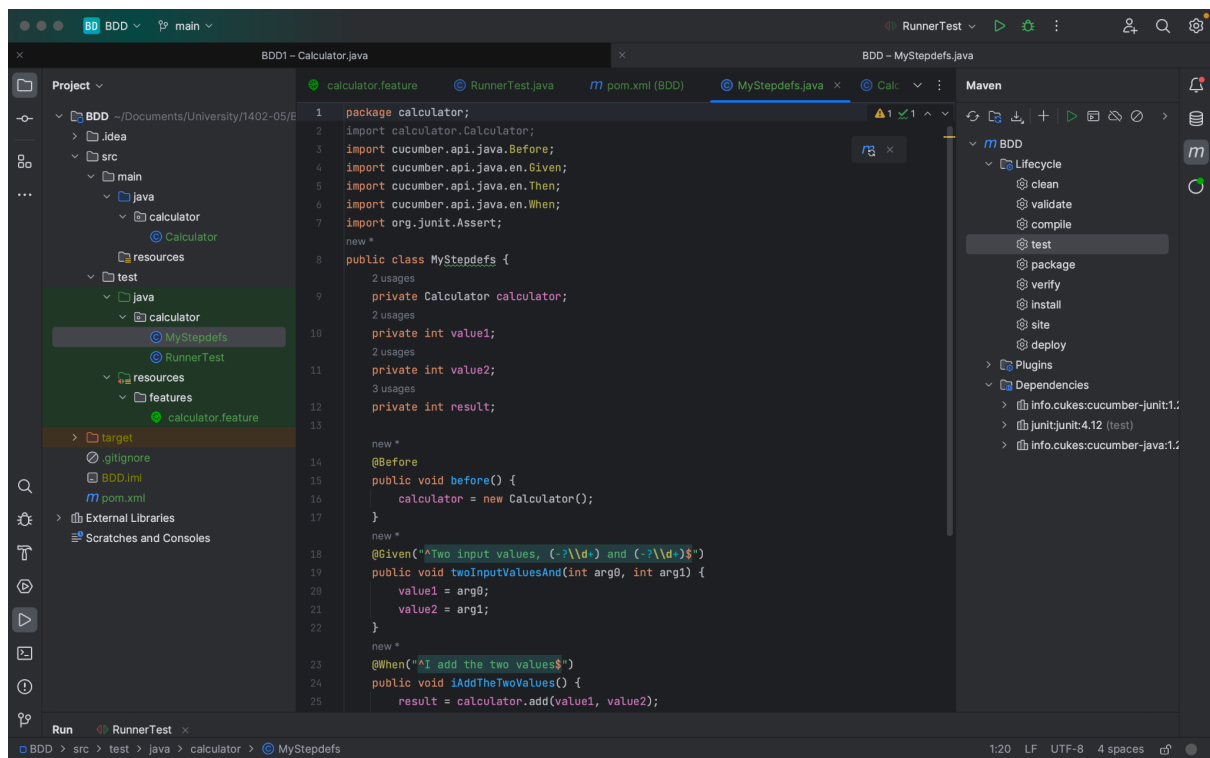


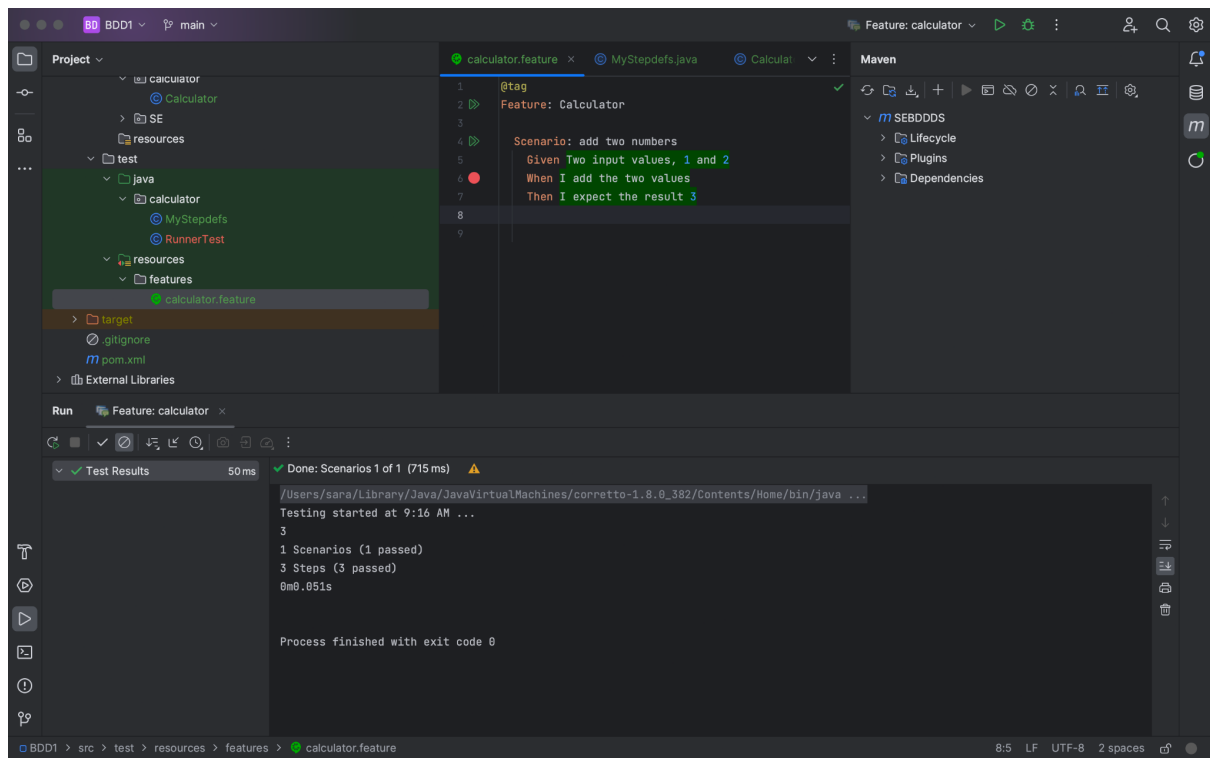


پس از آن در مسیر test/resources یک دایرکتوری با نام features ایجاد میکنیم و همینطور یک پکیج جدید با نام calculator در مسیر src/main/java ایجاد میکنیم. در دایرکتوری features هم یک فایل با نام calculator.feature ایجاد میکنیم.



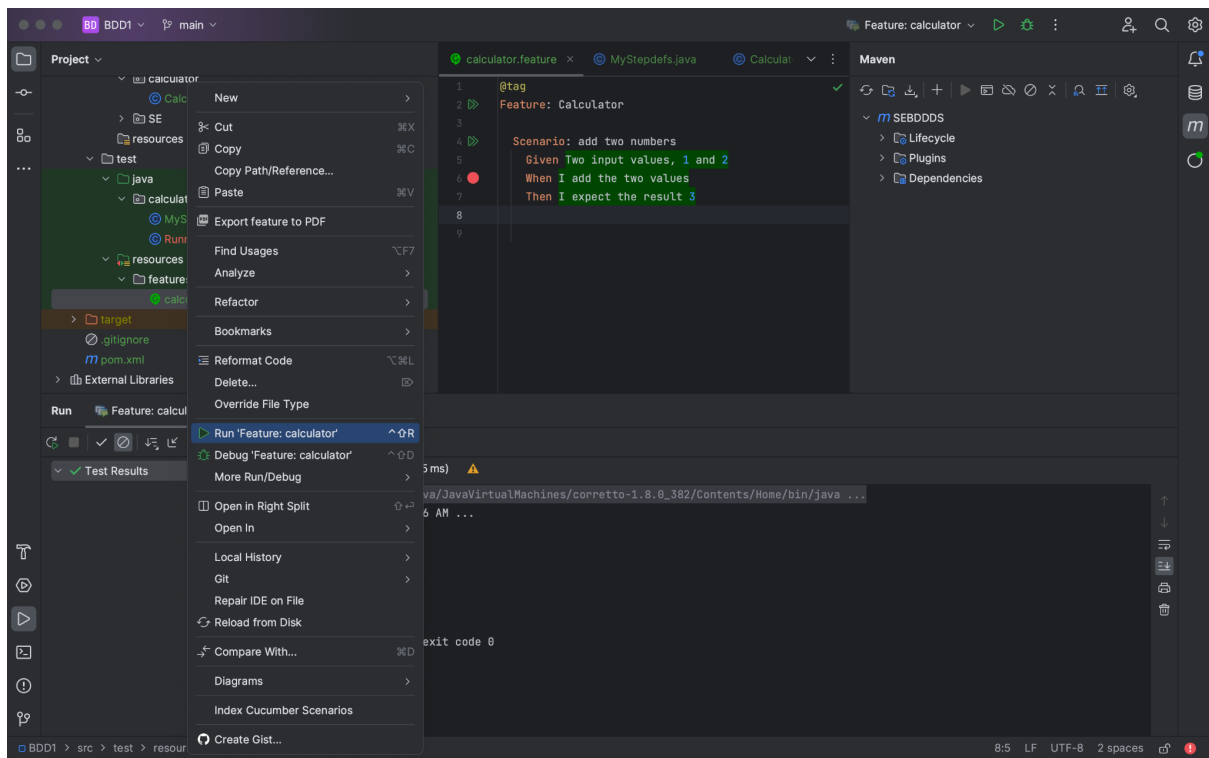
با زدن این گزینه در دایرکتوری calculator فایل با نام MyStepdefs ساخته میشود که توابع را آنجا پیاده سازی میکند. توابع را به صورت زیر تکمیل میکنیم.



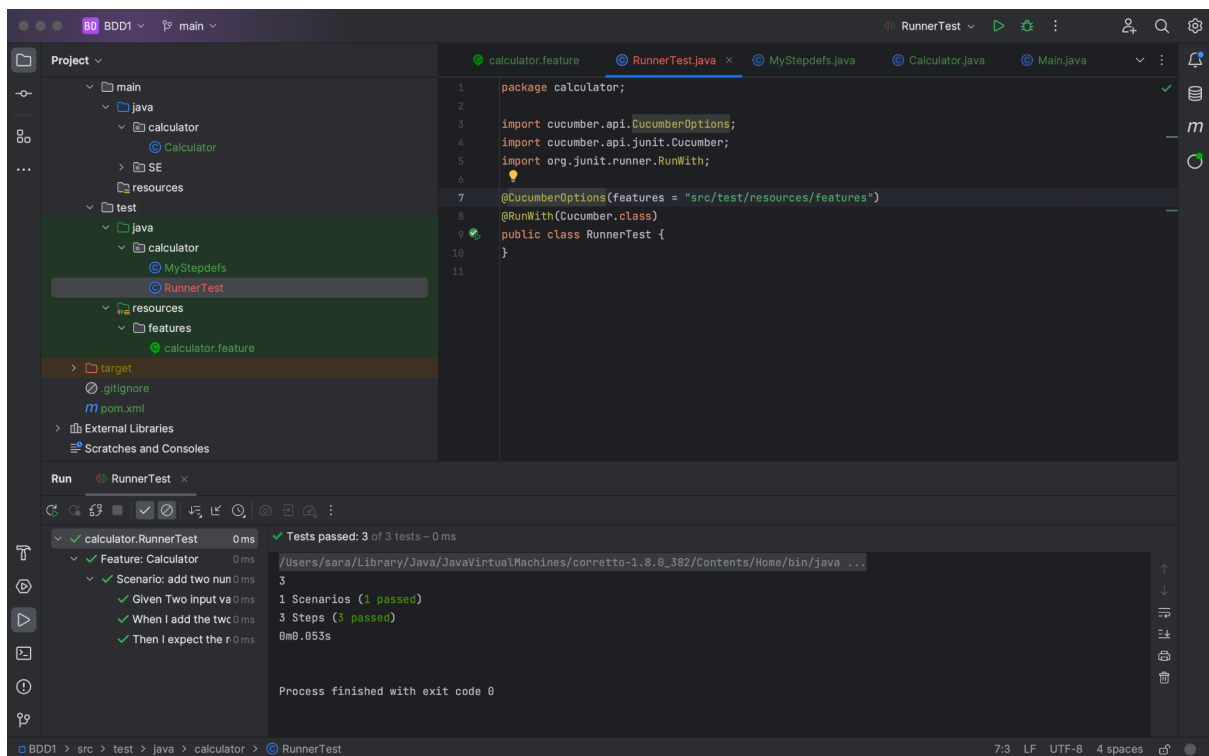


اکنون مطابق تصویر زیر روی calculator.feature کلیک راست کرده و گزینه run را میزنیم.

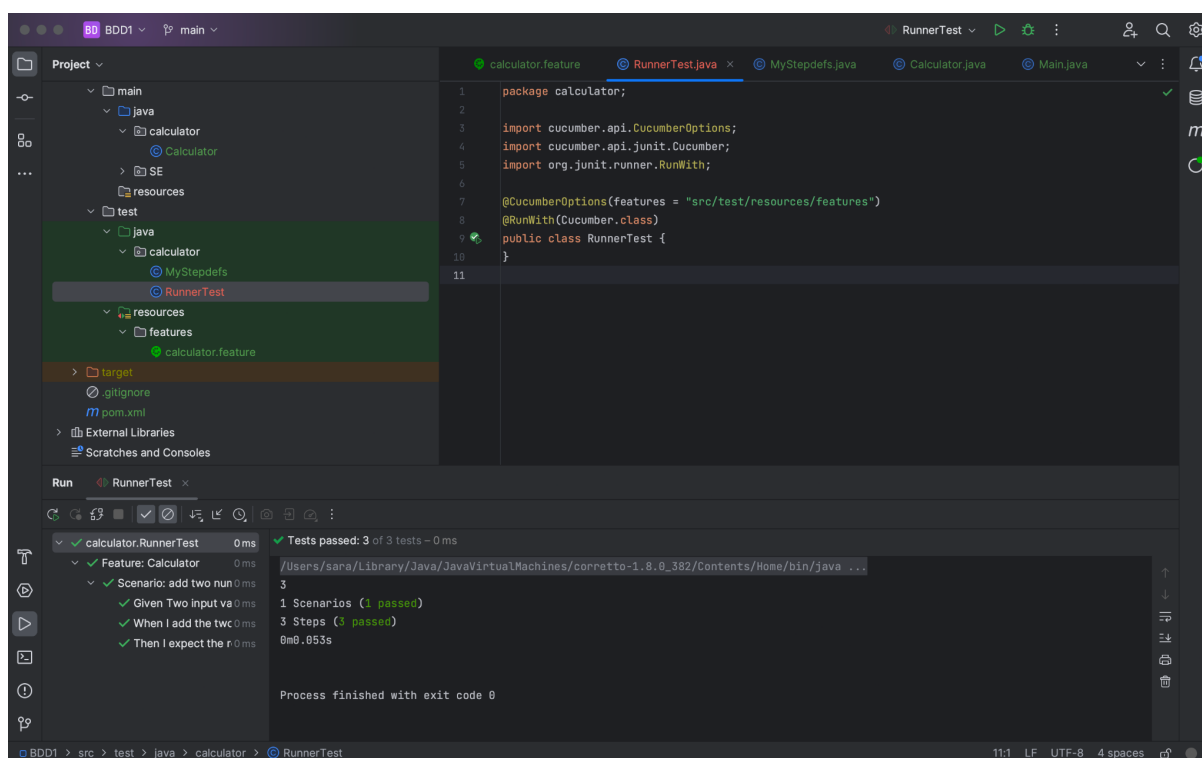
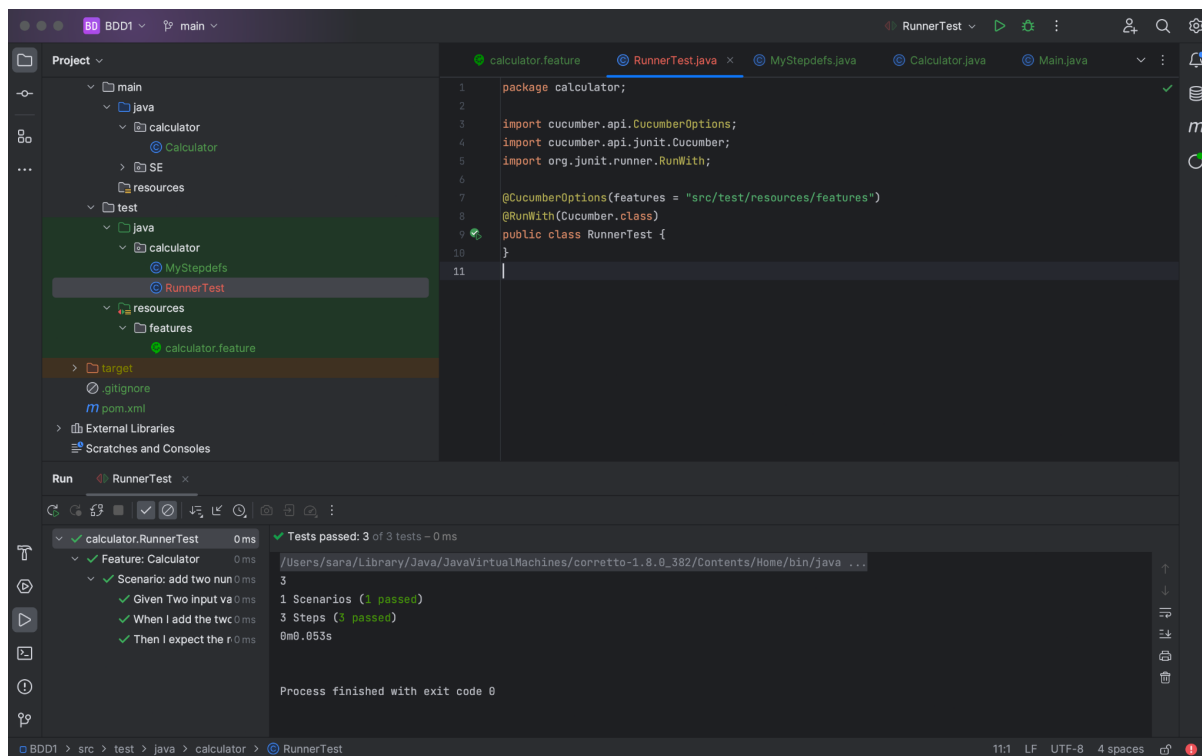




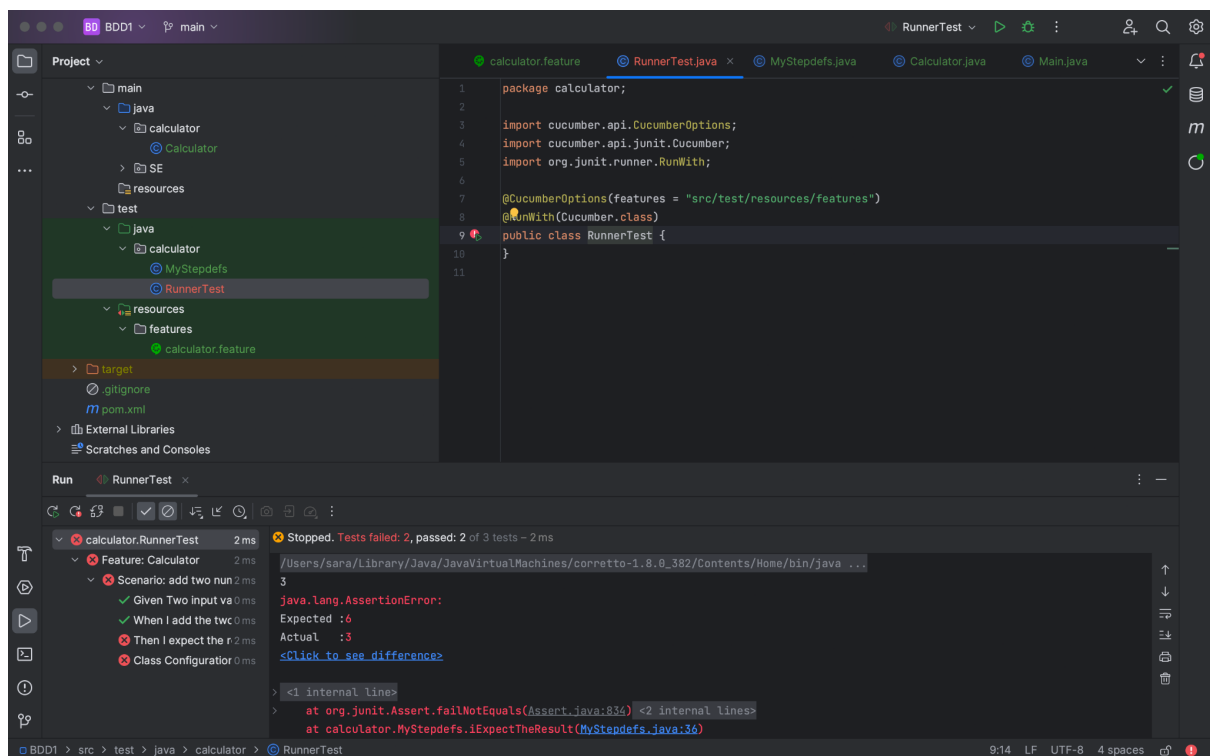
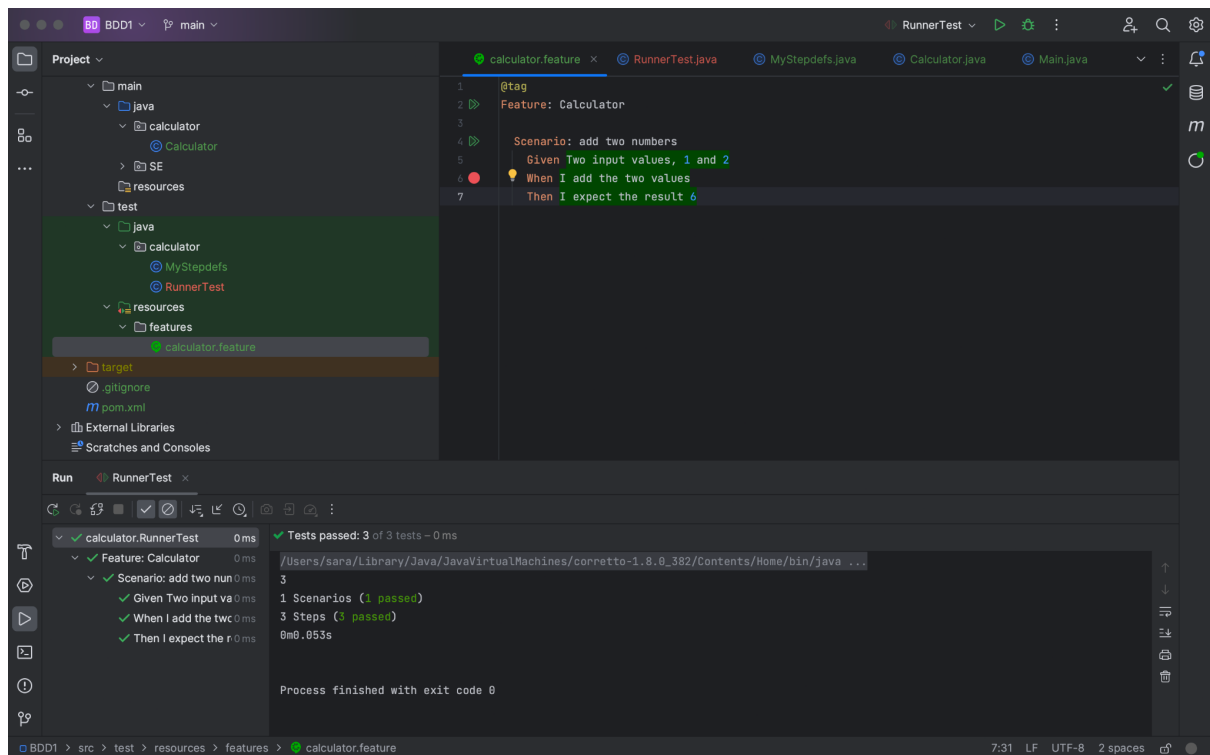
همانطور که مشاهده میکنید نتیجه به صورت زیر است.



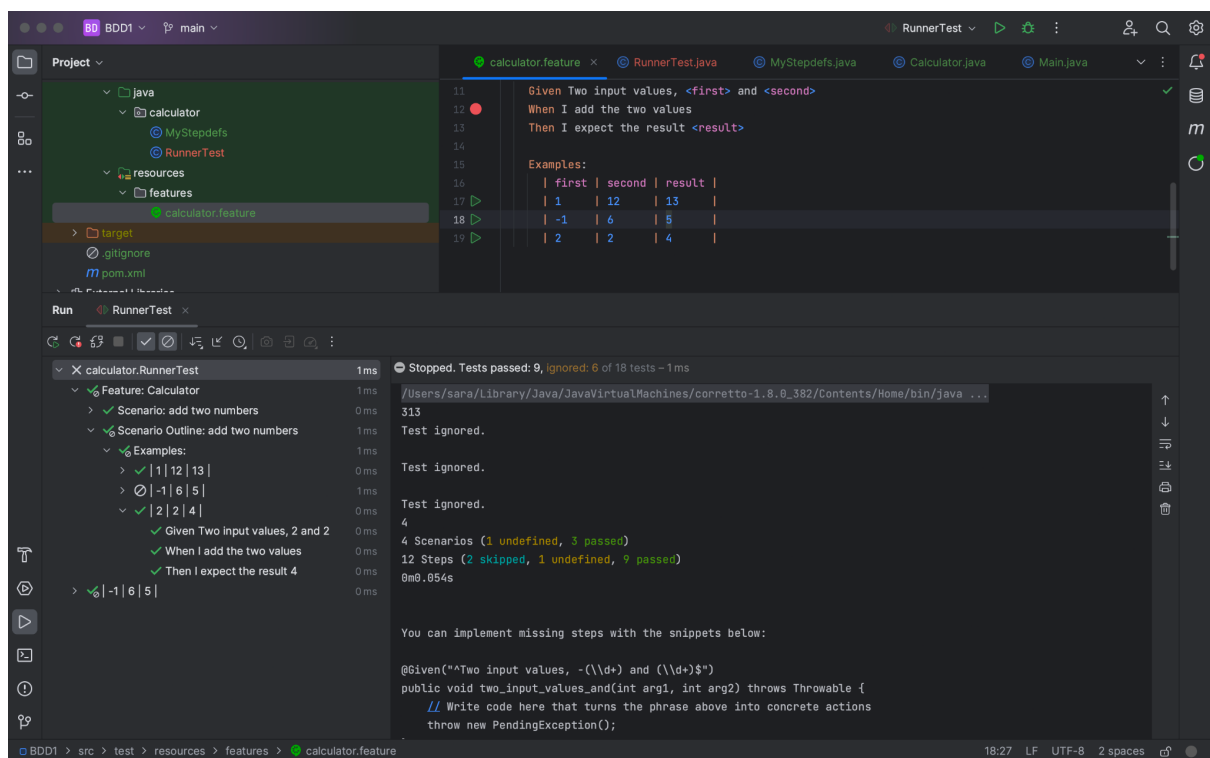
مطابق مراحل گفته شده در دستورکار RunnerTest را در پکیج calculator ایجاد میکنیم و آن را اجرا میکنیم.



حال اعداد تست را تغییر میدهیم تا نتیجه را ببینیم.

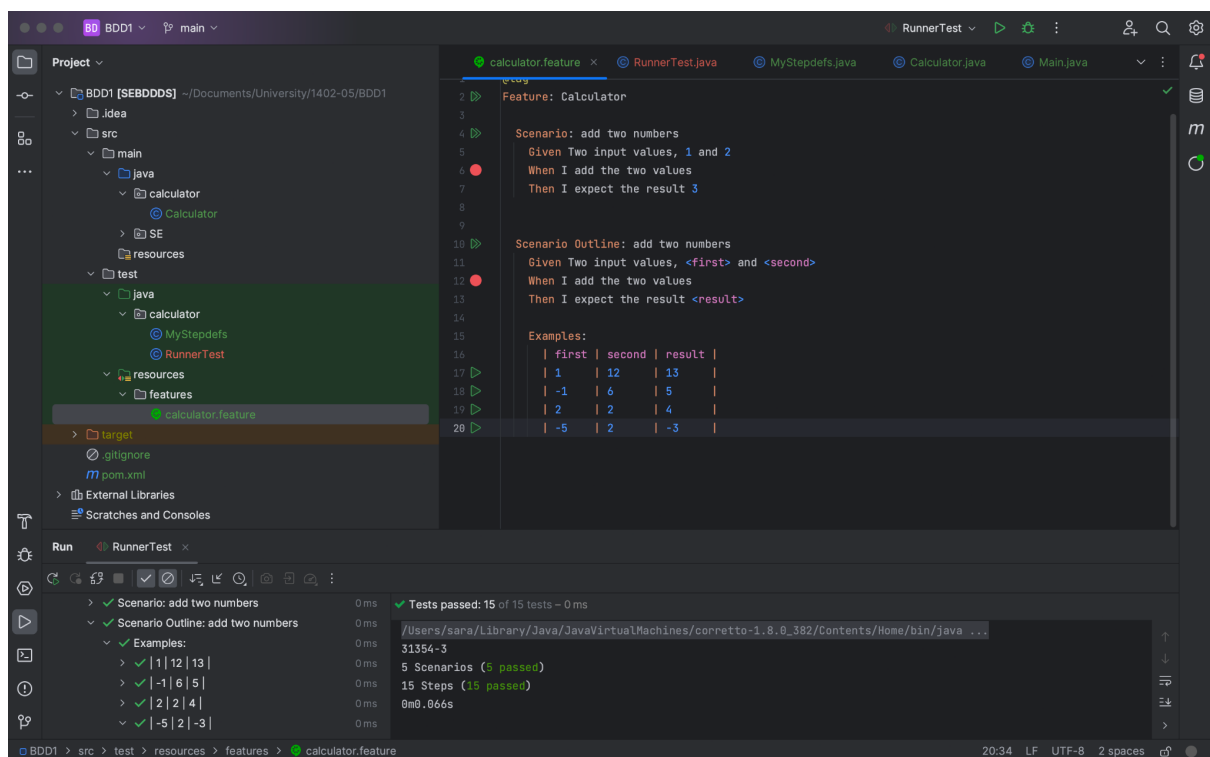


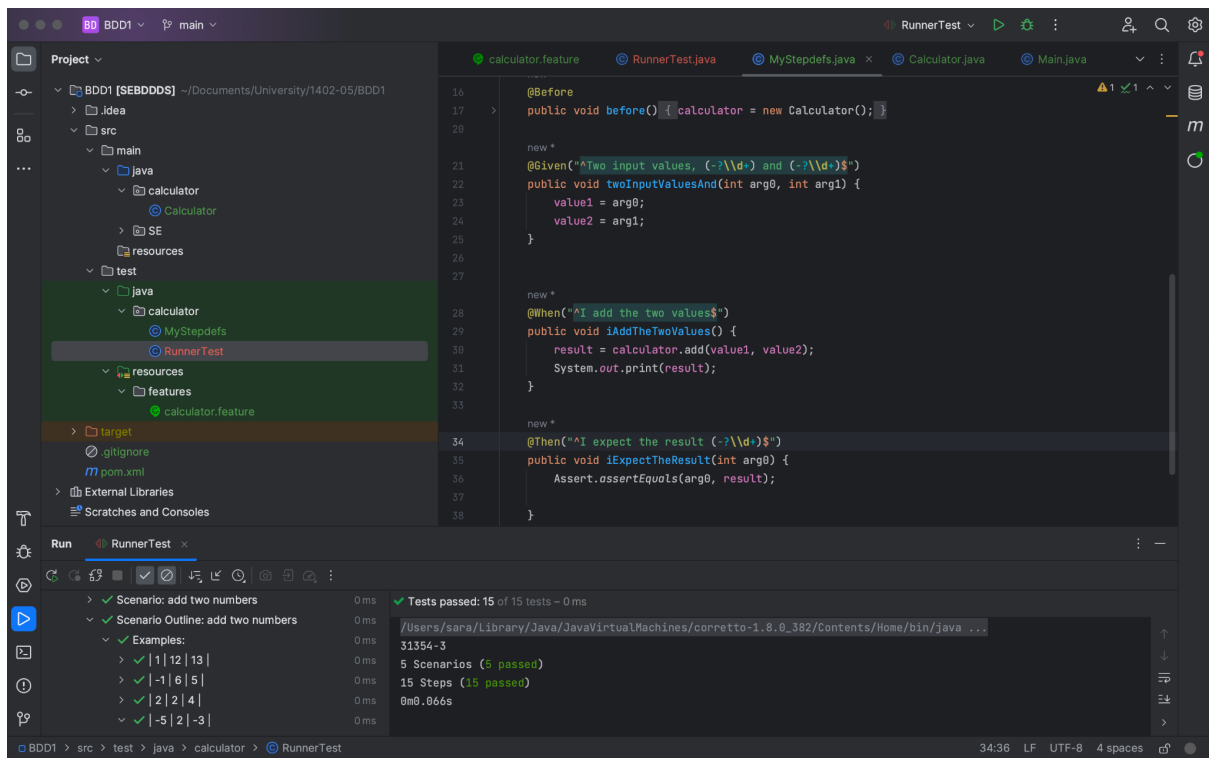
حال سناریوی outline را طبق دستور کار آزمایش به فایل feature اضافه میکنیم. و دوباره اجرا میکنیم. نتیجه آزمایش را در شکل زیر مشاهده میکنید.



تستی که عدد منفی دارد همانطور که مشاهده میکنید اجرا نشده است. علت آن است که ما اعداد منفی را در نظر نگرفته بودیم. برای اصلاح آن در فایل MyStepdefs در رجکس اعداد، اعداد منفی را هم در نظر میگیریم:

(تست آخر را خودمان نوشتیم)





```
@Given("^Two input values, (-?\d+) and (-?\d+)$")
```

همانطور که مشاهده میکنید این مشکل رفع شده است و علت آن در نظر نگرفتن اعداد منفی بوده است.