گزارش آزمایش دوم

آز مهندسی نرم افزار

محسن دهقان‌کار - ۹۷۱۰۵۹۲۸

سجاد ریحانی - ۹۷۱۰۵۹۷۴

توضیحات آزمایش

آدرس گیت پروژه:

<https://github.com/sreyhani/SoftwareEnginneringLab-TDD>

گام اول

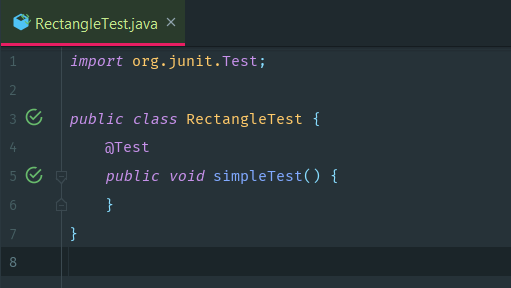
ابتدا پروژه maven جدیدی با IntelliJ ایجاد می‌کنیم.

وابستگی junit را به فایل pom.xml اضافه می کنیم.



سپس فایل RecatngleTest.java  را برای شروع نوشتن تست ها در مسیر src/test/java ایجاد می‌کنیم.

و در آن یک تست خالی برای کلاس RectangleTest می‌نویسیم و آن را اجرا می‌کنیم تا ببینیم کار می‌کند یا نه



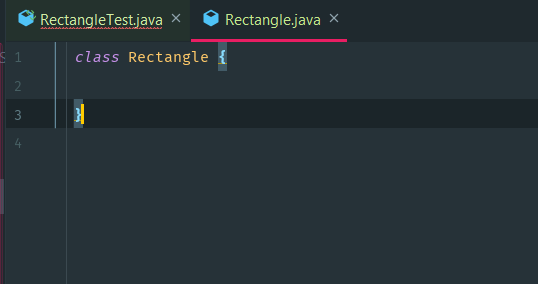
بعد اولین تست را برای ساختن یک شی از کلاس Rectangle می نویسیم



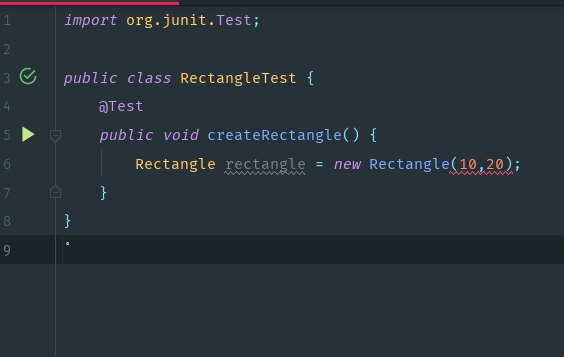
این تست کامپایل نمیشود چون کلاس Rectangle وجود ندارد.

حالا فایل Rectangle.java را در مسیر src/main/java می سازیم و در آن کلاس Rectangle را تعریف می‌کنیم.

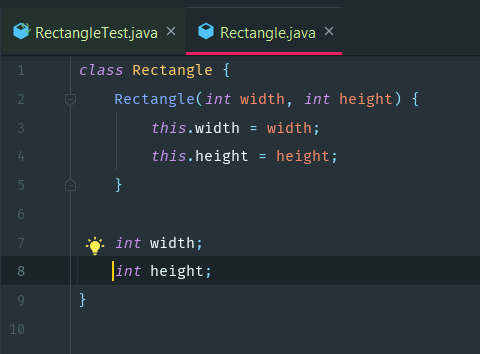
حالا تست اجرا می‌شود.



حالا در تست ورودی ارتفاع و عرض را به کلاس Rectangle میدهیم. دوباره کامپایل نمیشود چون constructor مناسب برا آن ندارد



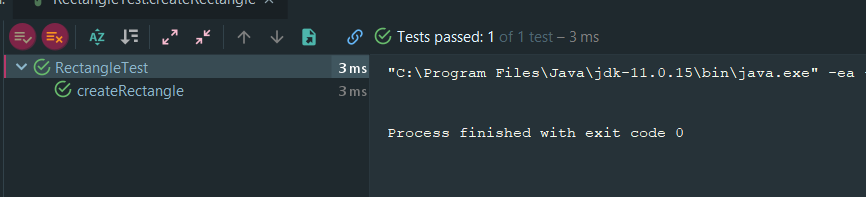
به کلاس Rectangel ارتفاع و عرض را اضافه می کنیم و سازنده مناسب برای آن را ایجاد می‌کنیم





حالا تست میکنیم مقادیر درست ست شده باشند





حالا تست مربوط به حساب کردن مساحت را ایجاد میکنیم



چون این متد نیست کار نمیکند

متد را اضافه می‌کنیم

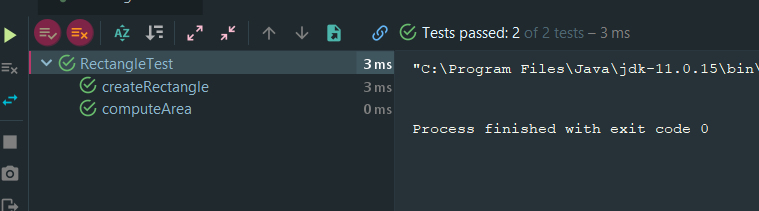


حالا تست کامپایل میشود ولی پاس نمیشود.

برای پاس شدن باید منطق را نیز در تابع پیاده کنیم



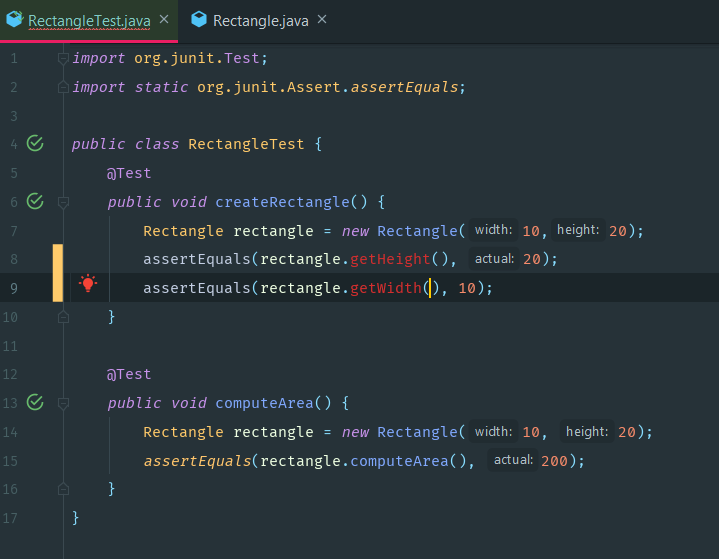
حالا تست پاس می‌شود



موارد ایجاد شده را کامیت میکنیم

گام دوم

ابتدا در تست createRectangle تغییر ایجاد میکنیم که مقادیر را با تابع get بخواند.



حالا تغییرات لازم در کلاس را انجام میدهیم

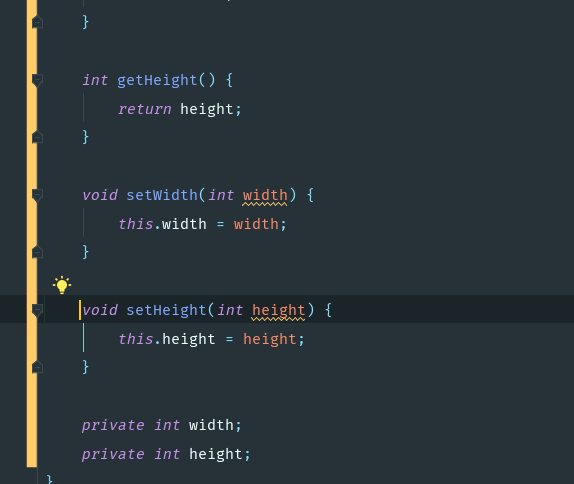


تست پاس میشود

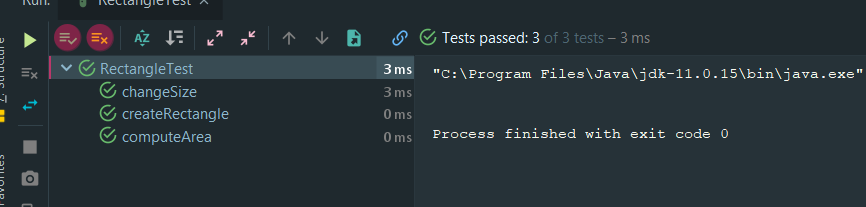
تست جدیدی برای تغییر مقادیر مستطیل مینویسیم



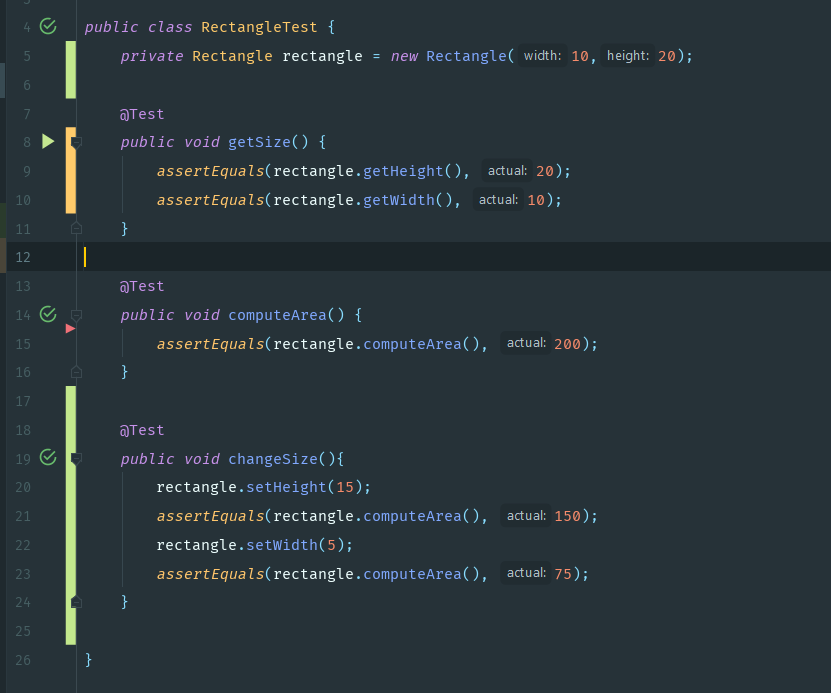
توابع را پیاده سازی میکنیم



تست ها پاس میشوند



حالا یک refactor انجام میدهیم و مستطیلی که در هر سه تست ساخته میشود را به عنوان پارامتر در کلاس تست میبریم که تکرار نشود

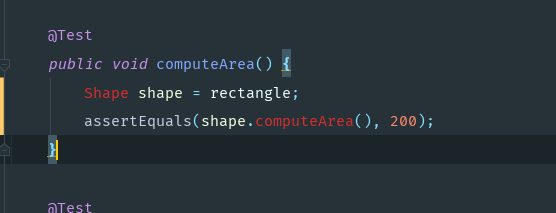


گام سوم

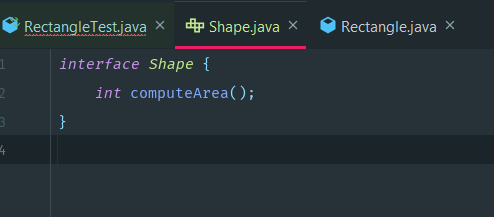
برای اضافه کردن مربع، یک گزینه این است که مربع را بچه کلاس مستطیل در نظر بگیریم و بگوییم که مربع نوعی مستطیل است. اما این مدلسازی غلط است چرا که مربع یک محدودیت و پیش شرط اضافه نسبت به مستطیل دارد که عرض و ارتفاعش با هم برابر باشند و این اصل LSP را نقض میکند و نمیتوان از آن در همه جا به جای پدرش استفاده کرد.

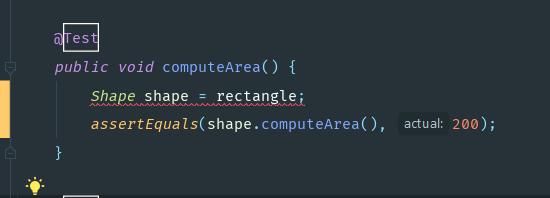
راه حل بهتر این است که یک اینترفیس به اسم Shape داشته باشیم که یک تابع computeArea دارد و کلاس مستطیل و مربع این اینترفیس را پیاده سازی میکنند.

ابتدا تست مساحت مستطیل را تغییر میدهیم



اینترفیس Shape را میسازیم با متد computeArea

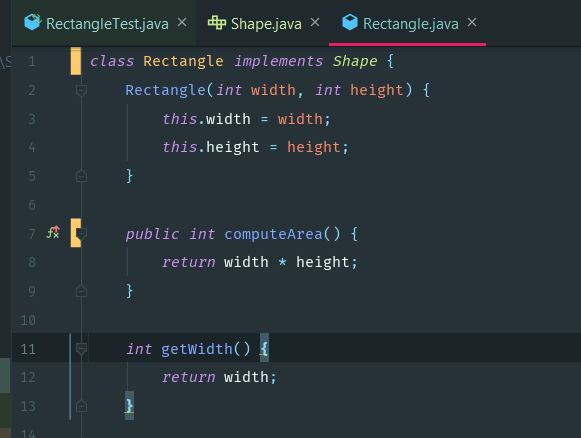


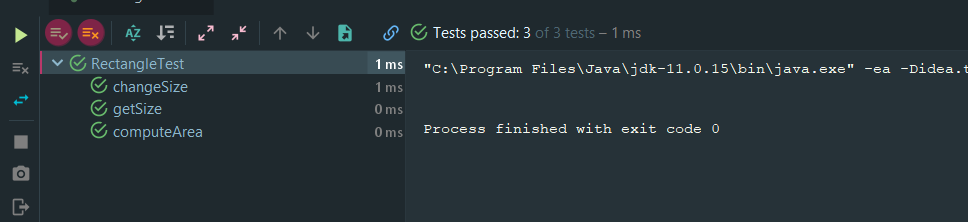


حالا خطا میدهد که تایپ مستطیل مناسب shape نیست

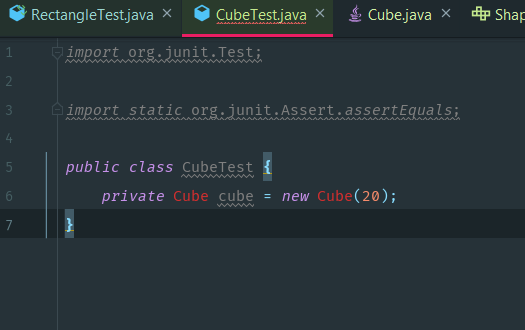
در کلاس مستطیل اینترفیس را به ارث میبریم و متد را اوراید میکنیم

سی





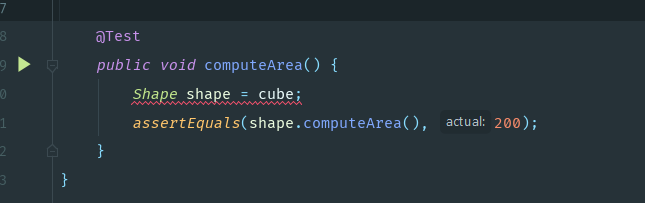
حالا برای تست مربع یک فایل جدید میسازیم به اسم CubeTest



کلاس Cube را میسازیم



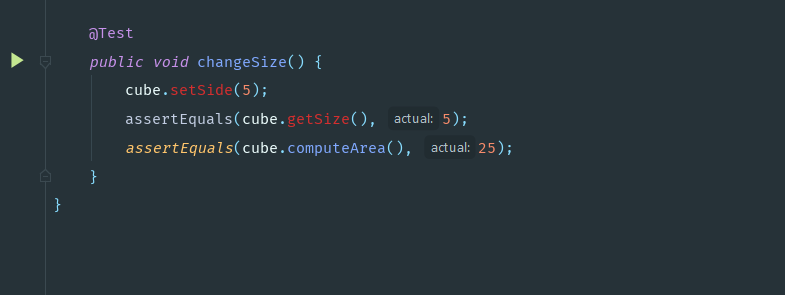
تست مربوط به محاسبه مساحت را اضافه میکنیم



در کلاس Cube اینترفیس را پیاده سازی میکنیم



تست تغییر ضلع مربع را اضافه میکنیم



پیاده سازی مکنیم ستر و گتر رو

