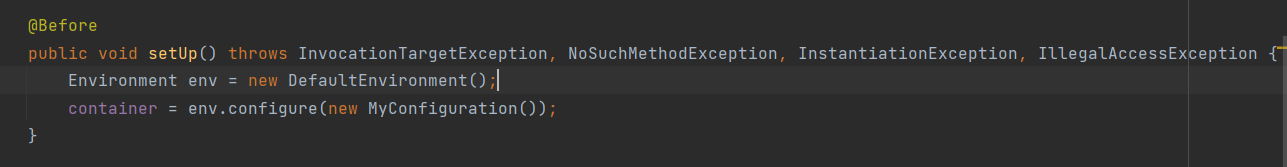
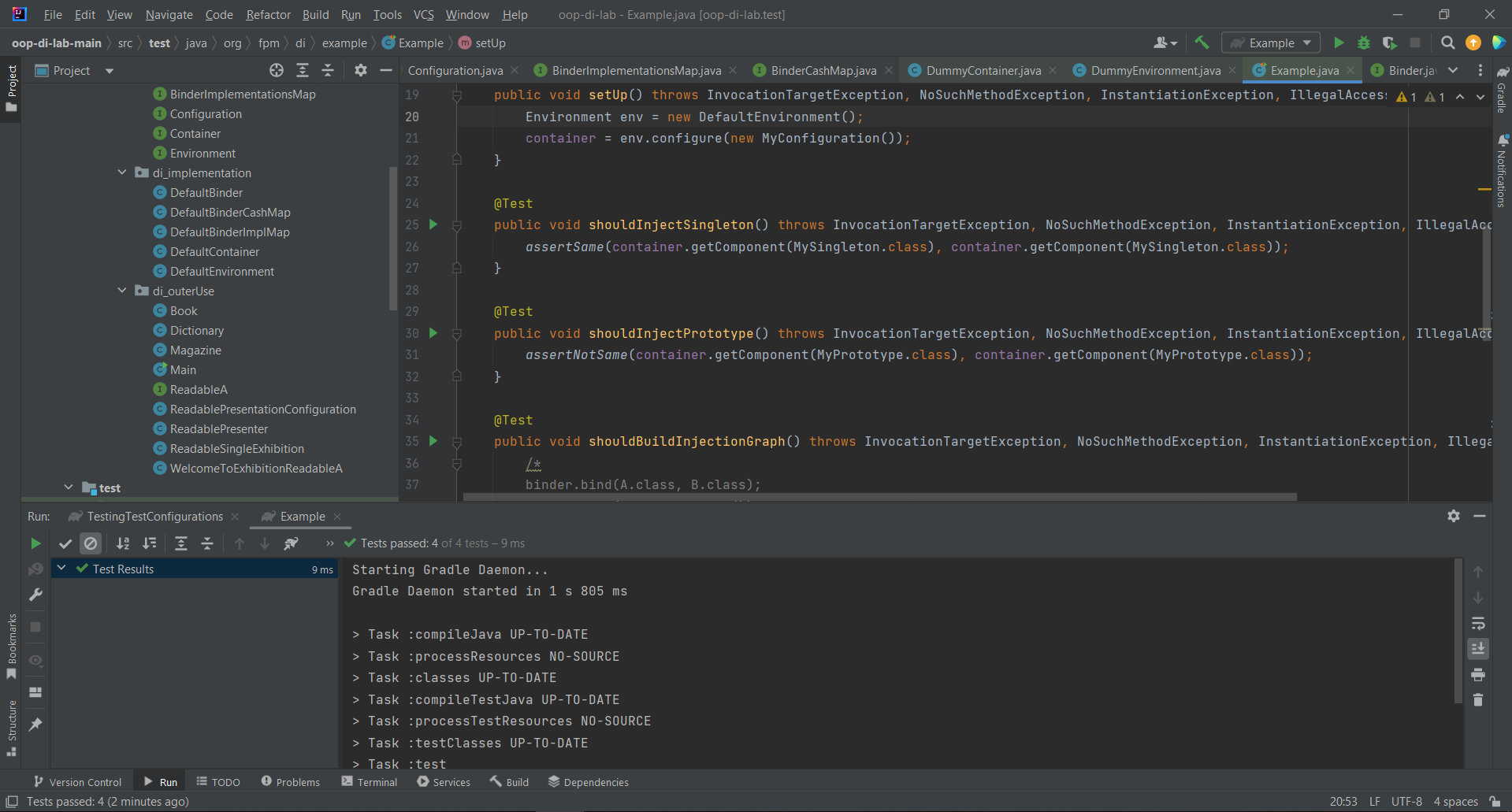
1 Частина. Тестування в Example.java

При тестуванні покриваються класи DefaultEnvironment.java, MyConfiguration.java (знаходиться в test/java/ org.fpm.di.example, відповідно, покриваються всі класи з цього пакету), DefaultContainer.java, DefaultBinder.java.



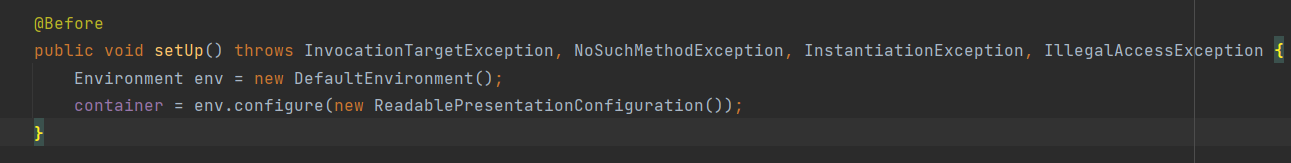
В кожному з тестів фігурує тестування лише одного методу container-а - getComponent(). Але “за кулісами” програми в класі DefaultBinder.java виконується важливий для виконання container.getComponent() код.

Відповідно, із застосуванням Default\* імплементації бібліотеки (з включенням допоміжних классів мап DefaultBinderCashMap, DefaultBinderImplMap чиї інтерфейси додані до бібліотеки) мають місце такі результати тестування:

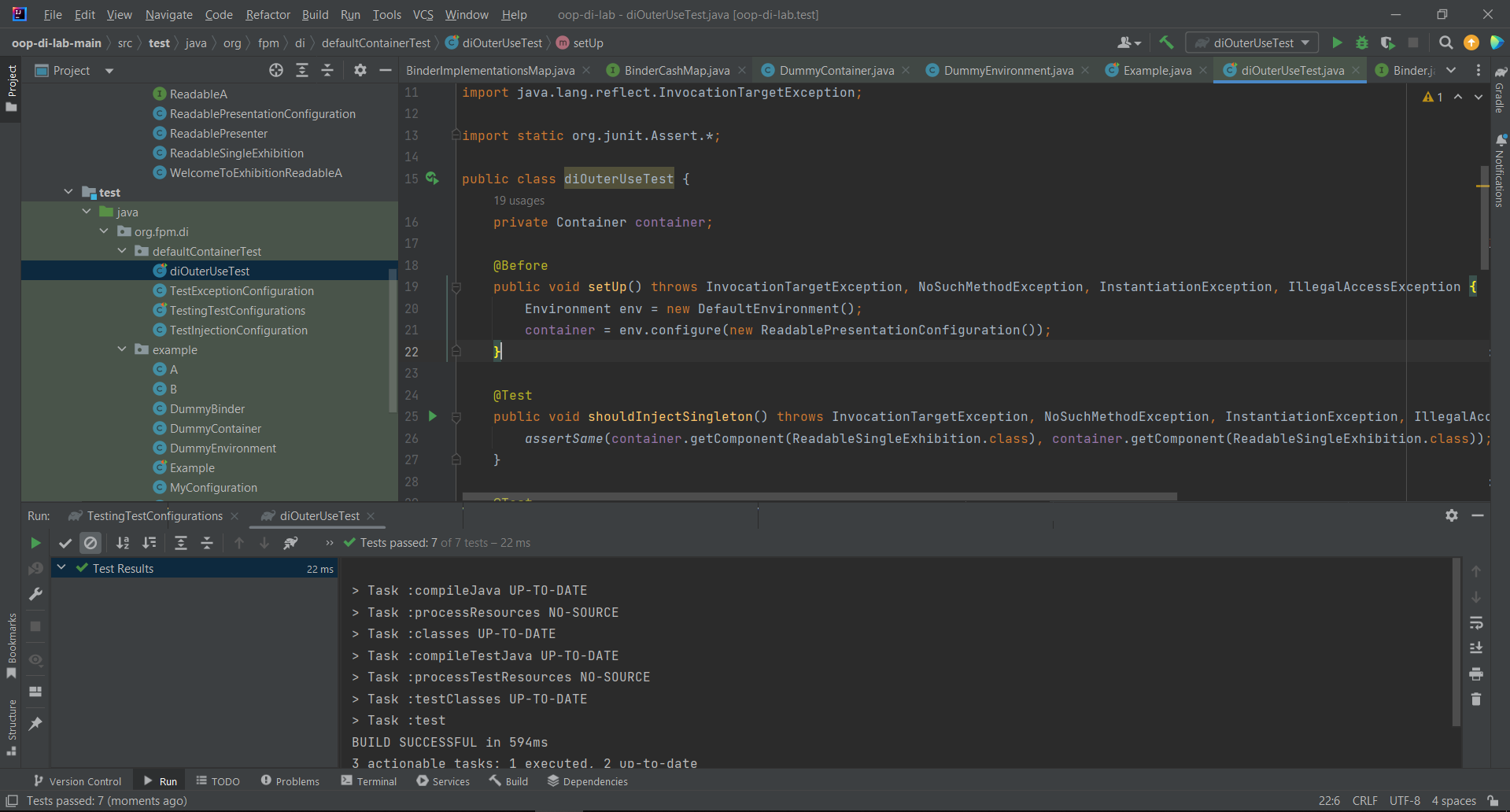


2 Частина. Тестування в diOuterUseTest.java

При тестуванні покриваються класи DefaultEnvironment.java, ReadablePresentationConfiguration.java (знаходиться в main/java/ org.fpm.di\_outerUse, відповідно, покривається частина класів з цього пакету), DefaultContainer.java, DefaultBinder.java.

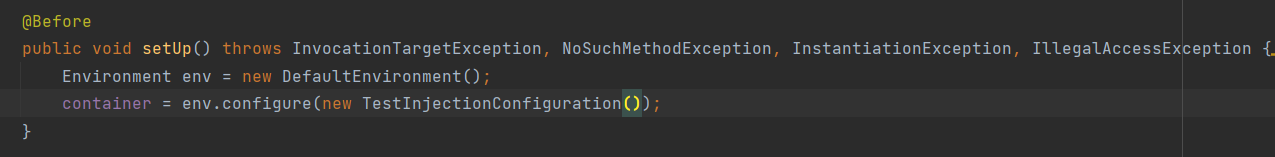


Подальший процес тестування відбувається аналогічно до 1 Частини (Example.java)

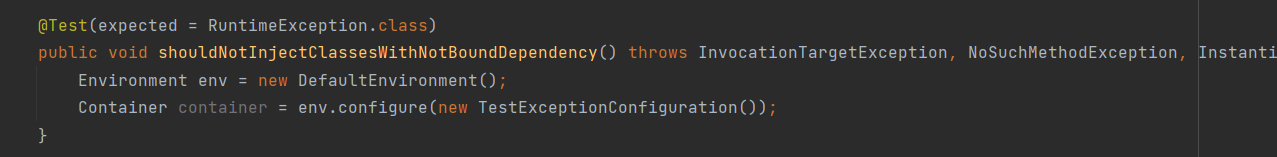


Частина 3. Тестування TestingTestConfiguration.java

При тестуванні покриваються класи DefaultEnvironment.java, TestExceptionConfiguration.java, TestInjectionConfiguration.java (знаходяться в test/java/ org.fpm.di.defaultContainerTest, покривається частина класів з цього пакету, а саме дві конфігурації), DefaultContainer.java, DefaultBinder.java.



Спочатку тестується, чи створиться контейнер з помилковою конфігурацією.



Потім тестується контейнер з конфігурацією зазначеною в методі setUp(), що анотований @Before.

