

Introducció a Concordion amb IntelliJ

1. Introducció

Concordion és un framework orientat al desenvolupament de tests d'acceptació. Funciona a través de tests escrits en HTML o mitjançant Markdown Language (a partir de la versió 2.0) que al seu torn són instrumentalitzats amb atributs especials que el framework interpreta per executar aquests tests.

En comptes de forçar que els Product Owners hagin d'especificar els requisits en un llenguatge amb una determinada estructura, Concordion permet definir-los utilitzant llenguatge natural, podent utilitzar paràgrafs, taules, i signes de puntuació. Això fa que les especificacions siguin molt més llegibles i senzilles d'escriure, a més d'ajudar enormement a la compressió i acceptació del que se suposa ha de fer la funcionalitat que anem a provar.

2. Fent servir servir Concordion amb IntelliJ

2.1 – Conceptes clau

Cada característica o comportament s'ha d'especificar, implementar i verificar mitjançant les especificacions actives i la seva connexió amb el sistema en desenvolupament. Una especificació activa a Concordion consisteix en dues parts clau:

1. Un document de requisits ben escrit que descriu la funcionalitat desitjada (HTML). Les especificacions HTML contenen descripcions de la funcionalitat il·lustrada amb exemples de prova d'acceptació. Les dades d'exemples es marquen mitjançant etiquetes HTML.
2. Les proves d'acceptació s'escriuen en Java i s'anomena codi de fixació o «fixture code». Les proves es codifiquen implementant una extensió de Concordion d'un cas estàndard de Junit test case. El codi de fixació troba les dades d'exemple marcats per etiquetes i els utilitza per verificar el sistema en desenvolupament.

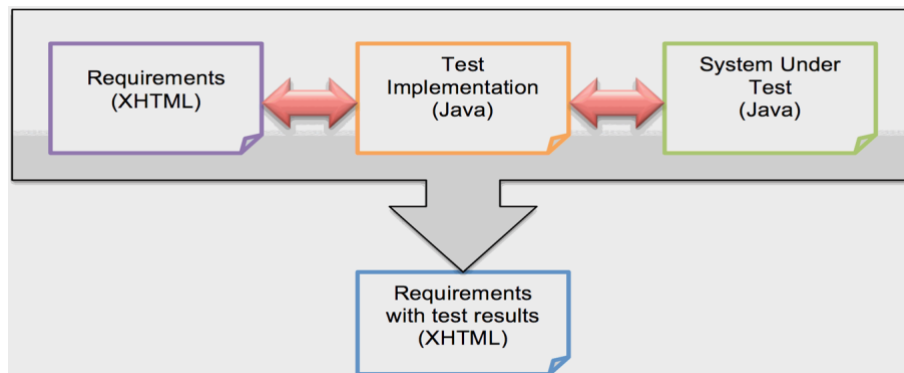


Figura 1: Concepte d'especificació activa.

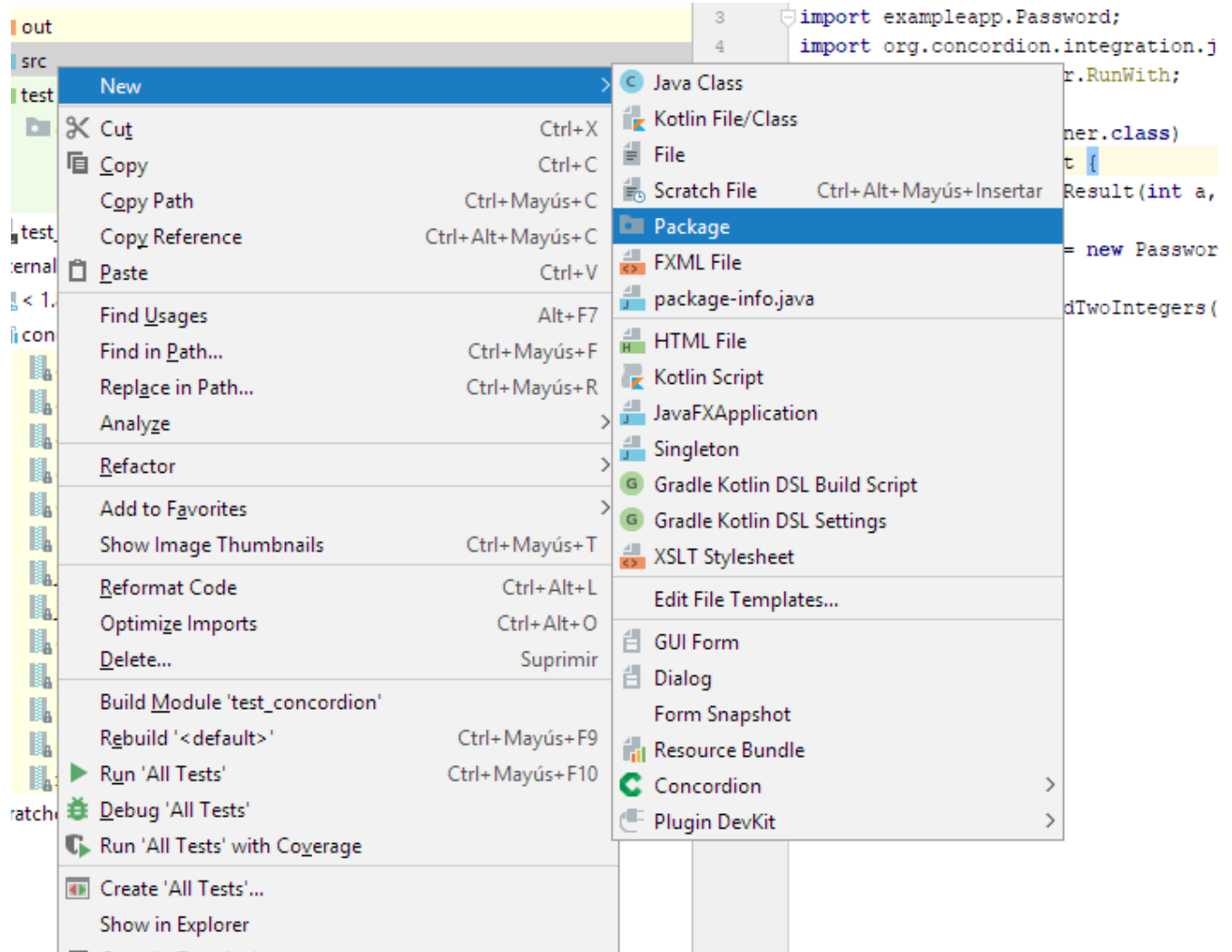
El concepte d'ús de Concordion s'il·lustra en el diagrama de la Figura 1. Les proves d'acceptació s'especifiquen utilitzant el llenguatge nadiu i s'organitzen en els fitxers Requirements (HTML). Les proves s'implementen en Java i connecten els exemples dels requisits amb el codi Java del sistema que s'està provant. El codi de prova està connectat amb els requisits mitjançant etiquetes HTML, que contenen ordres per Concordion. La realització de proves en Concordion dona com a resultat resultats de fitxers HTML que combinen els resultats de l'especificació i la prova originals. Les proves correctes es ressalten com a "verd" i no les que no tenen èxit amb "vermell". Aquest concepte de Concordion s'anomena especificacions actives (Active Specifications) a causa del fet que l'implementació de la prova vincula les especificacions amb el sistema. Qualsevol canvi en el sistema donarà lloc a proves fallides, el que ens recordarà que hem d'anar actualitzant les especificacions. D'aquesta manera les especificacions mai es faran obsoletes.

Per tal que Concordion funcioni, l'estructura de fitxers del vostre projecte ha de seguir les certes regles. En l'exemple que es mostra a la Figura 2 organitzem especificacions i provem els fitxers d'implementació en una estructura de paquets, aquí anomenada `exampleapp.spec`.

2.1.1 – Dummy example: Crear un projecte a IntelliJ amb Concordion.

- Crear un nou projecte amb IntelliJ de la mateixa forma que hem vist en el document de la sessió 1 per a crear projectes d'aquest IDE.

- Dins del directori src generem un nou Package (per exemple: ub.edu):

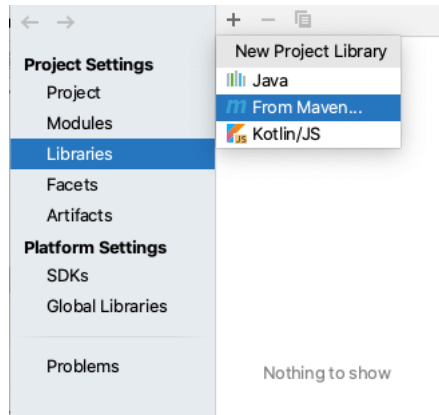


- Anem a fer el següent test d'acceptació:
En cas que [context] quan [event] el sistema [resultat]

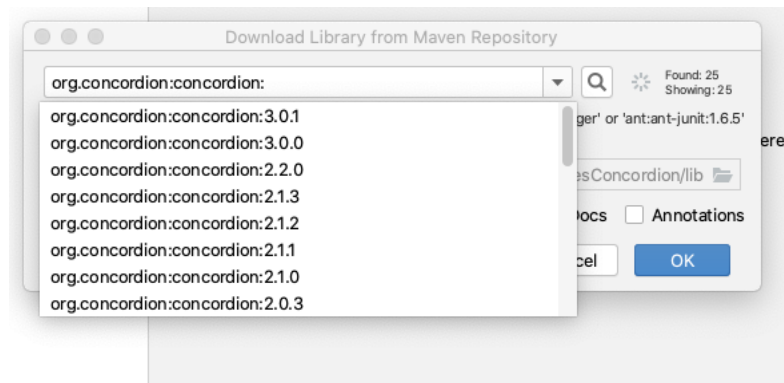
Test d'acceptació: US1.1 Suma de dos enters
En cas que l'usuari entri dos valors x i y
Quan premi <intro>
el sistema donarà com a resultat per pantalla la suma de x i y

Per això és necessari l'ús de Conordion

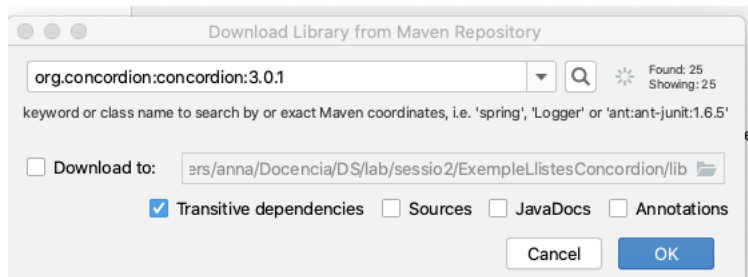
- Per a aconseguir que aquesta configuració es mantingui entre diferents instal·lacions, afegim el Concondion pel menú File->Project Structure -> Libraries posa l'opció + i selecciona Maven (i no Java com has fet abans).



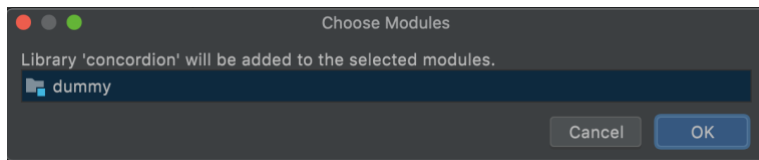
- Tecleja org.concondion:concondion: i dóna-li al botó de cerca



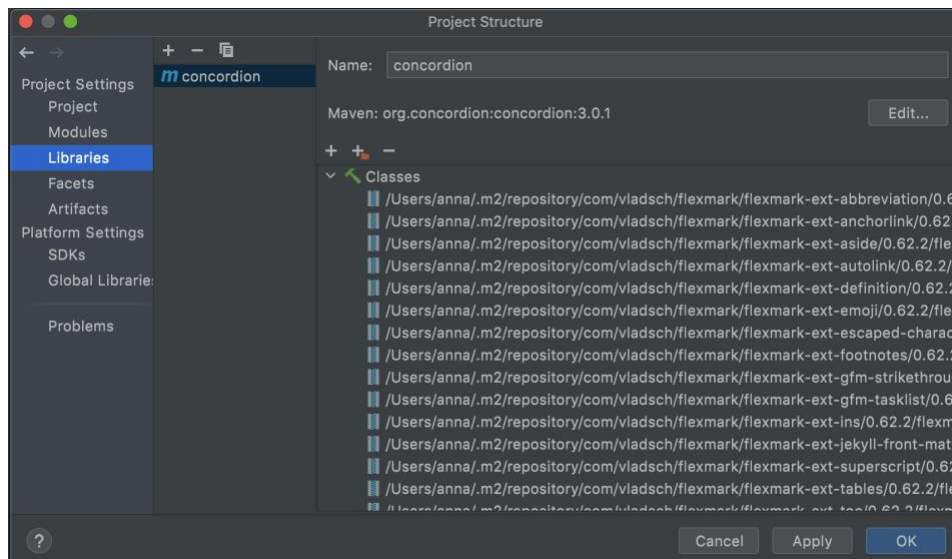
- Selecciona la versió 3.0.1



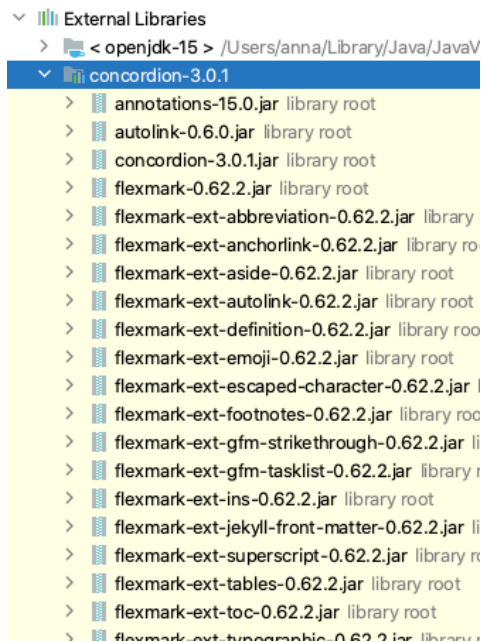
- A continuació et demanarà a quin mòdul vols afegir la llibreria:



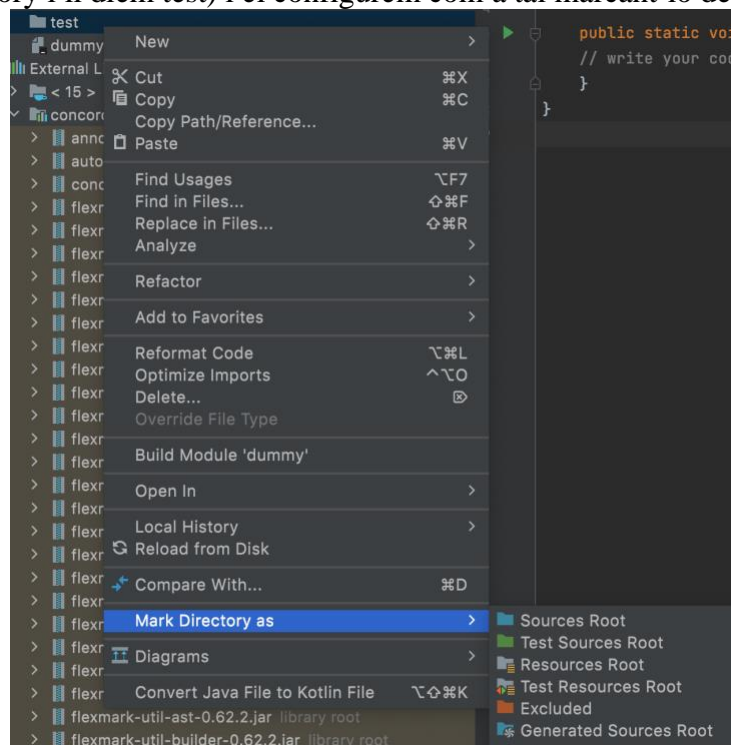
- Digues-li ok i així acabaràs tenint un projecte sense dependències de la carpeta on s'hagi instal·lat la llibreria
- Sortirà la finestra següent:



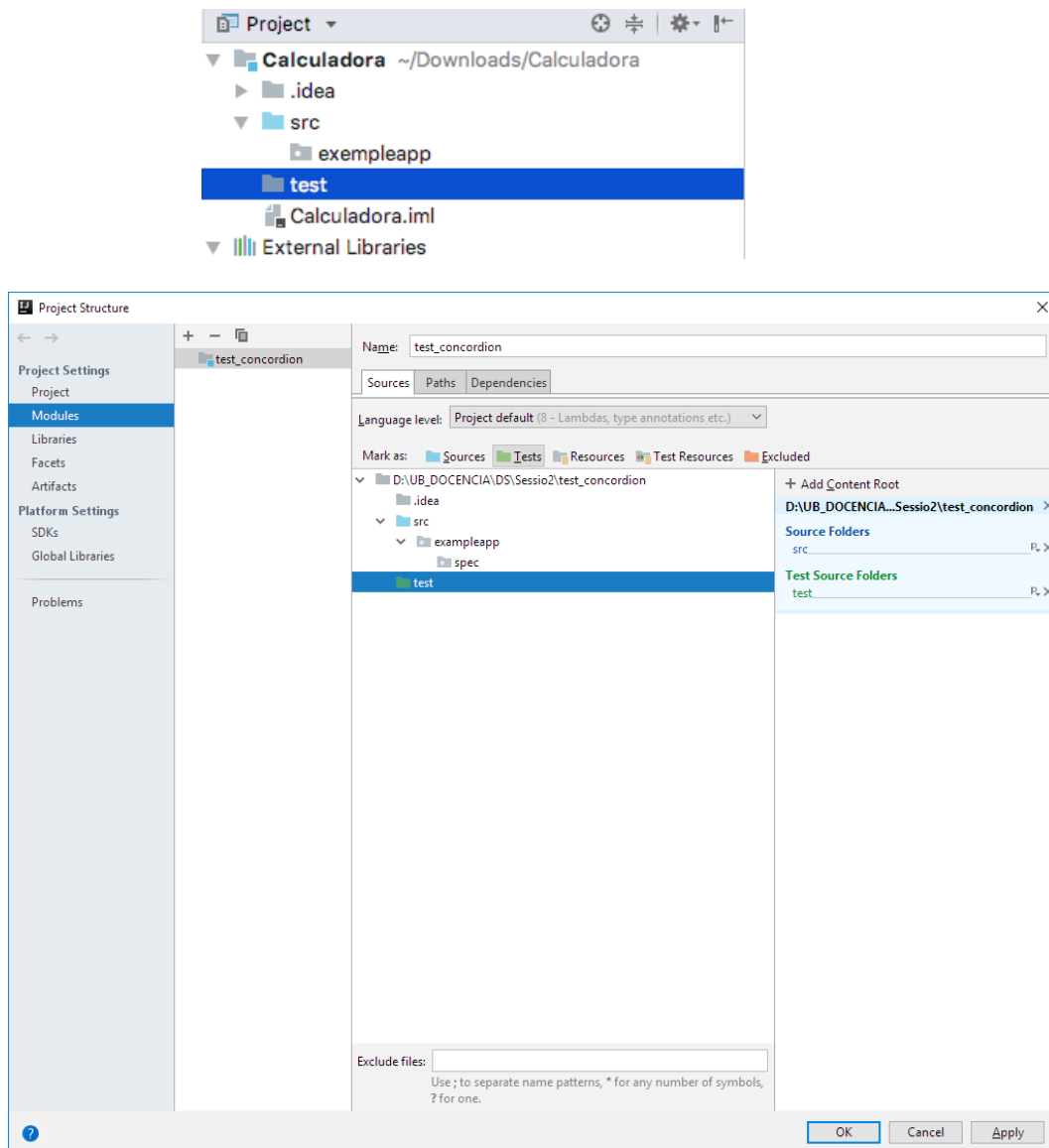
- Donem al botó d'OK. Si repassem la carpeta d'external libraries al projecte hauria d'aparèixer així:



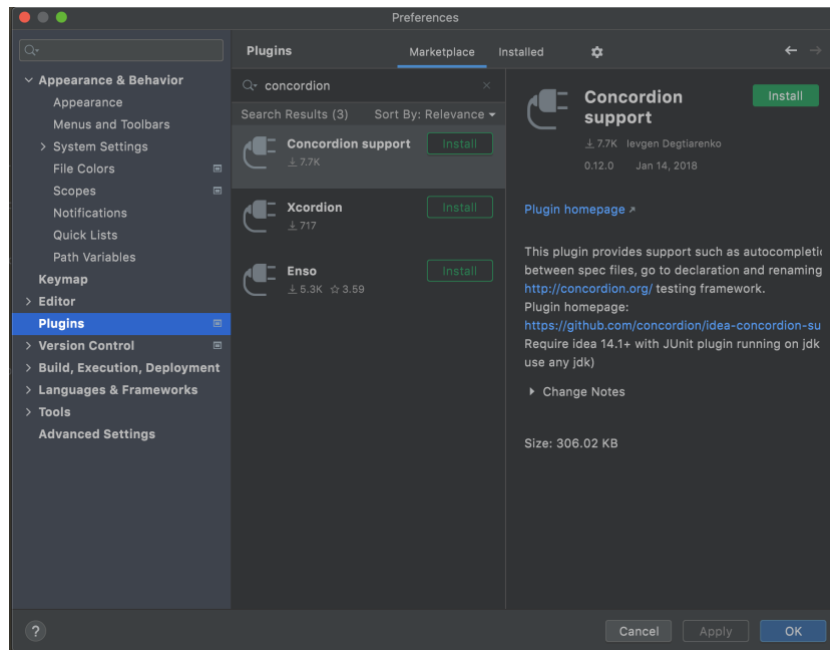
- A continuació, per crear el primer test, creem un directori de test (des del projecte, fem New-> Directory i li diem test) i el configurem com a tal marcant-lo de color verd.



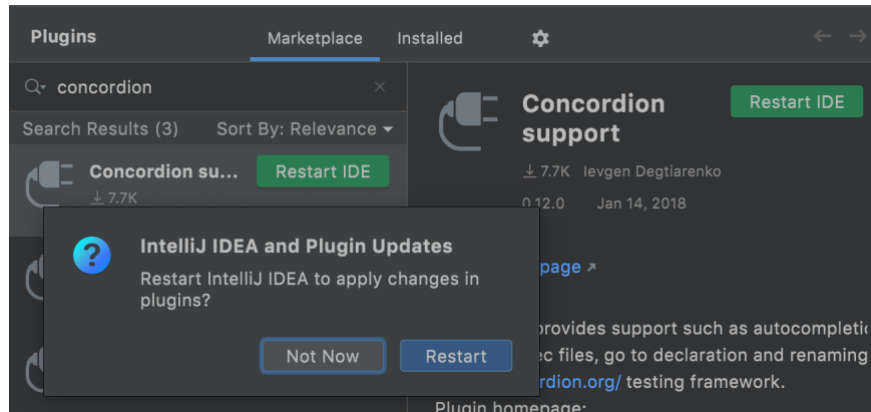
Es equivalent a fer-ho des del menú *Project structure* del menú *File*



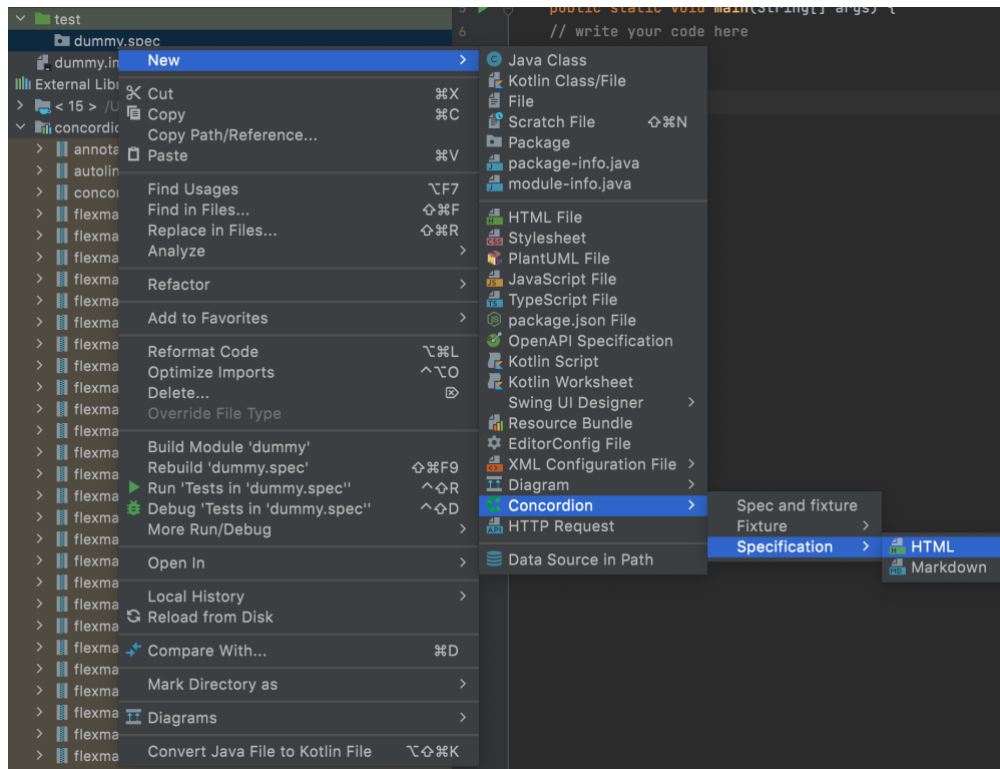
- Crearem un Package dins d'aquest directori (Per exemple: dummy.spec) i ens ajudarem del plug-in de Concordion per IntelliJ per a crear els *Fixture* i *Spec* a dins. Cal instal·lar aquest plugin des del menú *Preferences* dins del menú de *File* (o en instal·lacions de MaCMac des del menú *IntelliJ*). Seleccionar el submenú de *Plugins* i cercar el plugin "Concordion suport" al MarketPlace d'IntelliJ.



- Cal fer un Restart de l'IntelliJ:



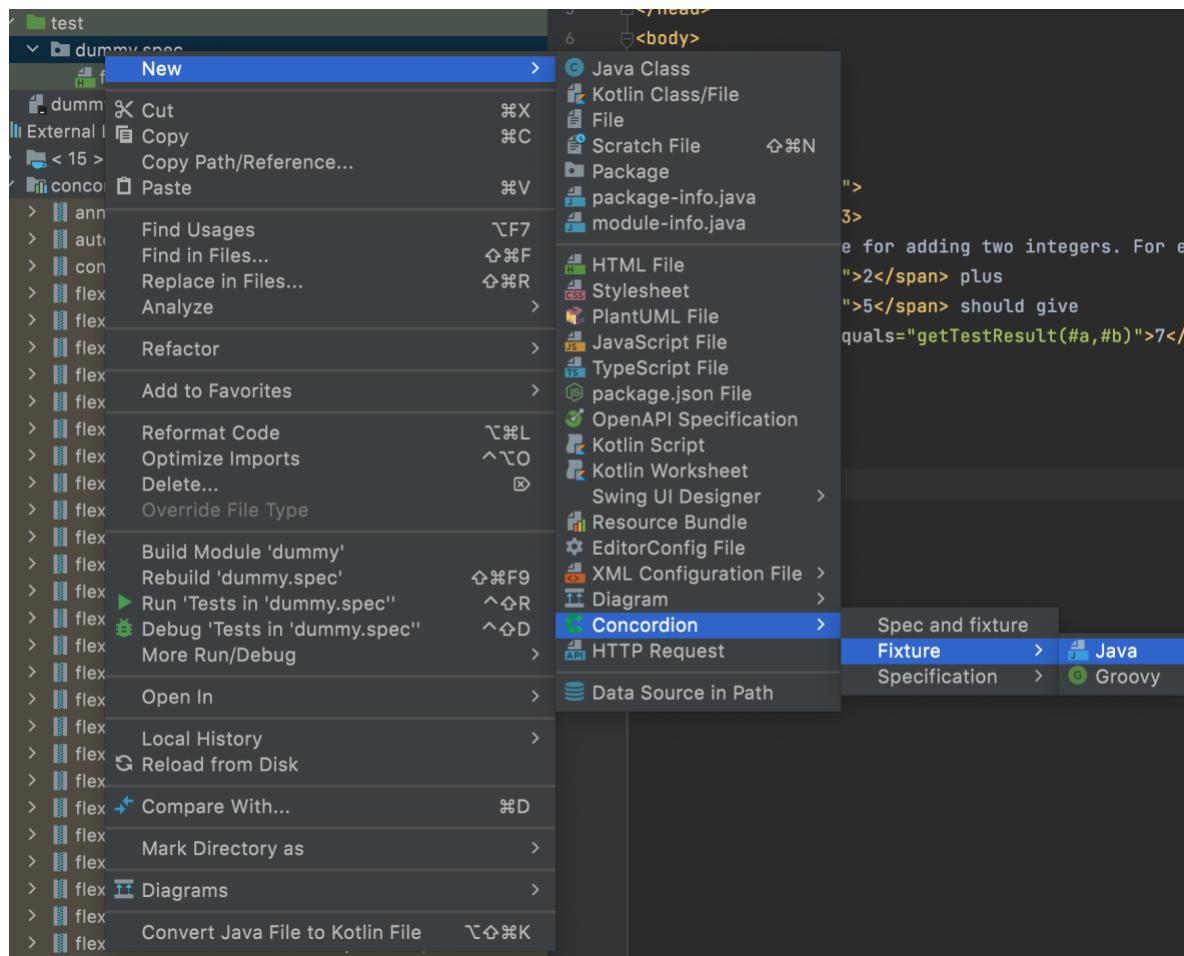
- Ara ja podem crear un test amb Concordion.



- Crearem primer un Specification anomenat firstUS.html que contindrà el següent codi:

```
<html xmlns:c="http://www.concordion.org/2007/concordion">
<head>
  <title>firstUS</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
</head>
<body>
<h1>firstUS</h1>
<p>
  Description
</p>
<div c:example="US1">
  <h3>Examples</h3>
  Provides feature for adding two integers. For example
  <span c:set="#a">2</span> plus
  <span c:set="#b">5</span> should give
  <span c:assertEquals="getTestResult(#a,#b)">7</span>.
</div>
</body>
</html>
```

- Finalment crearem el fixture firstUS.java per assegurar-nos que hem configurat Concordion correctament al projecte.

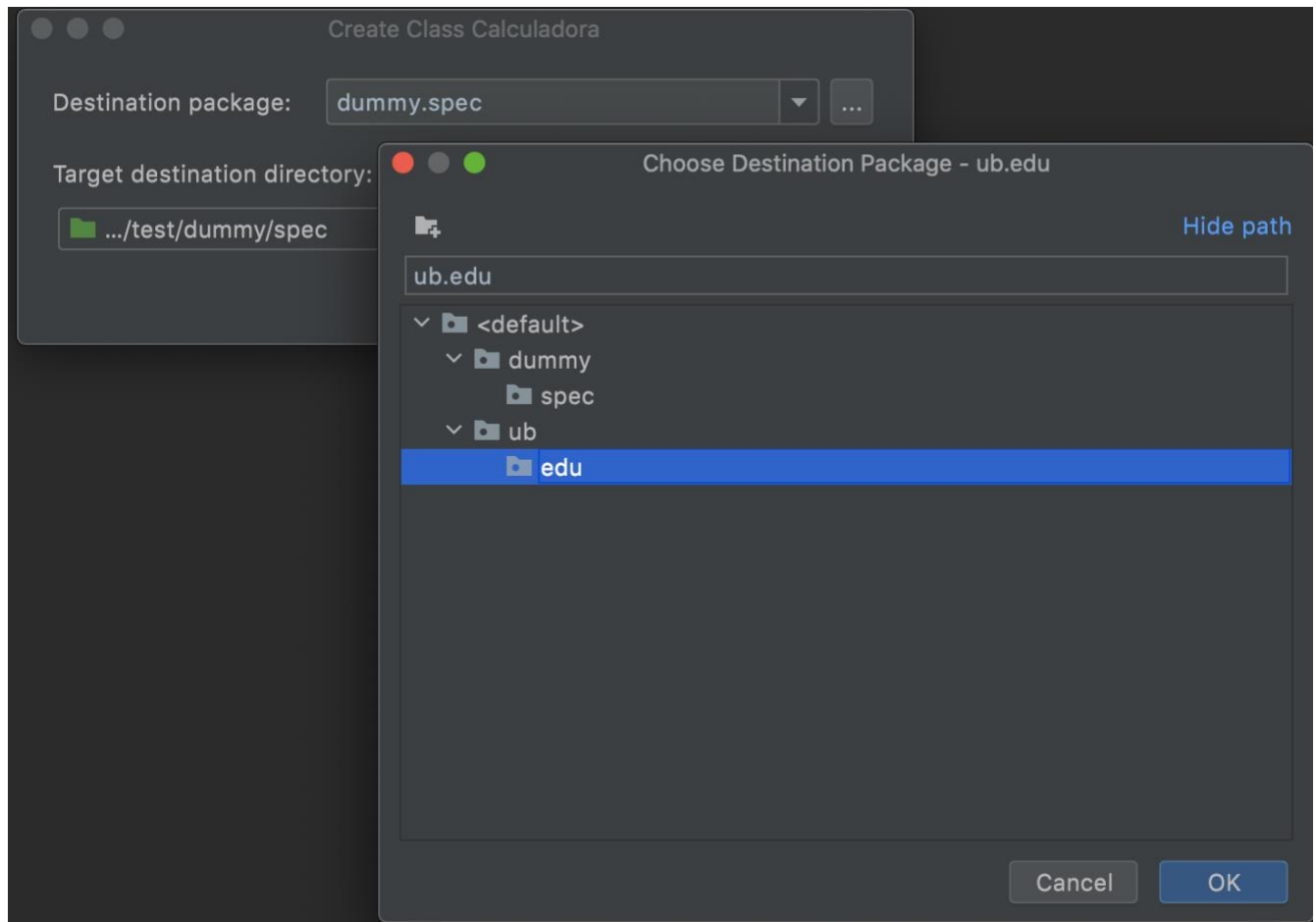


- El contingut del codi serà el següent:

```
import org.concordion.integration.junit4.ConcordionRunner;
import org.junit.runner.RunWith;

@RunWith(ConcordionRunner.class)
public class firstUS {
    public int getTestResult(int a, int b) {
        Calculadora calc = new Calculadora();
        return (calc.addTwoIntegers(a, b));
    }
}
```

- Ara cal posar la funcionalitat en la nostra aplicació de sumar dos enters. Fixeu-vos que si us poseu sobre la paraula `Calculadora` i premeu `Alt+Shift+Intro` us indica si voleu crear aquesta nova classe anomenada `Calculadora`. Ens demanarà on posar aquesta classe nova:

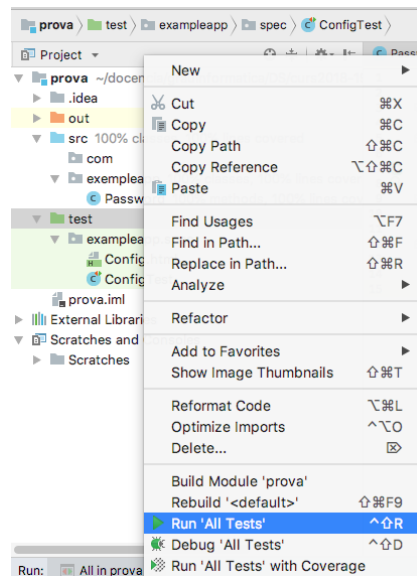


- En el package dels fonts (src) hem d'afegir el codi per a sumar dos enters:

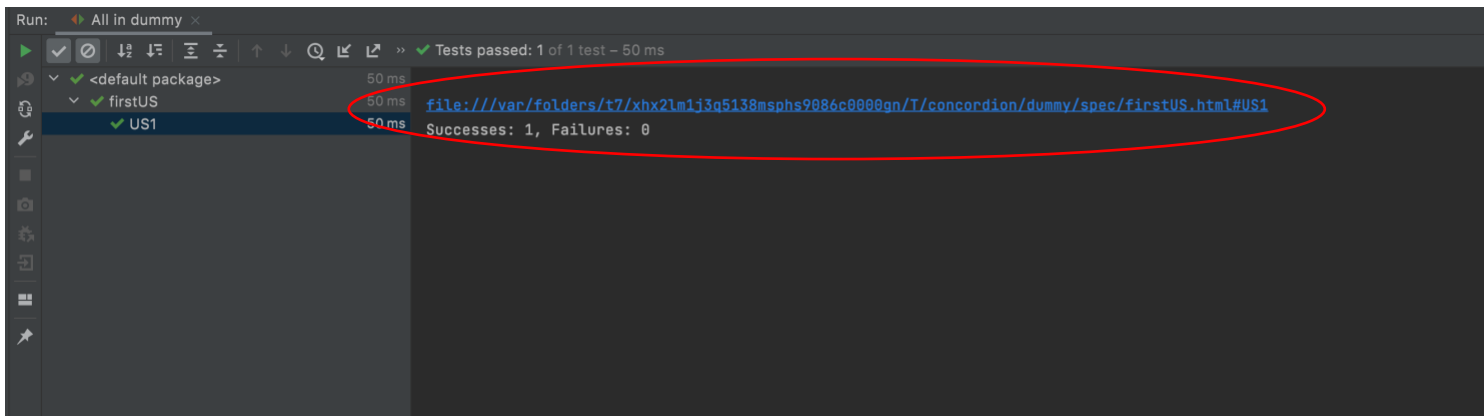
```
package ub.edu;

public class Calculadora {
    public int addTwoIntegers(int x, int y) {
        return x+y;
    }
}
```

- Tingueu en compte que us faran falta un sèrie d'imports per a que la classe compili, així també agafeu pràctica amb l'IDE i les funcions d'autoemplenar. Per executar els tests podeu seleccionar:



- El resultat al córrer els test serà:



Si obrim l'enllaç html que ens donen els test, podem veure'ls en format html en un navegador.

En aquest cas:

<file:///var/folders/t7/xhx2lm1j3q5138msphs9086c0000gn/T/concordion/dummy/spec/firstUS.html#US1>

firstUS

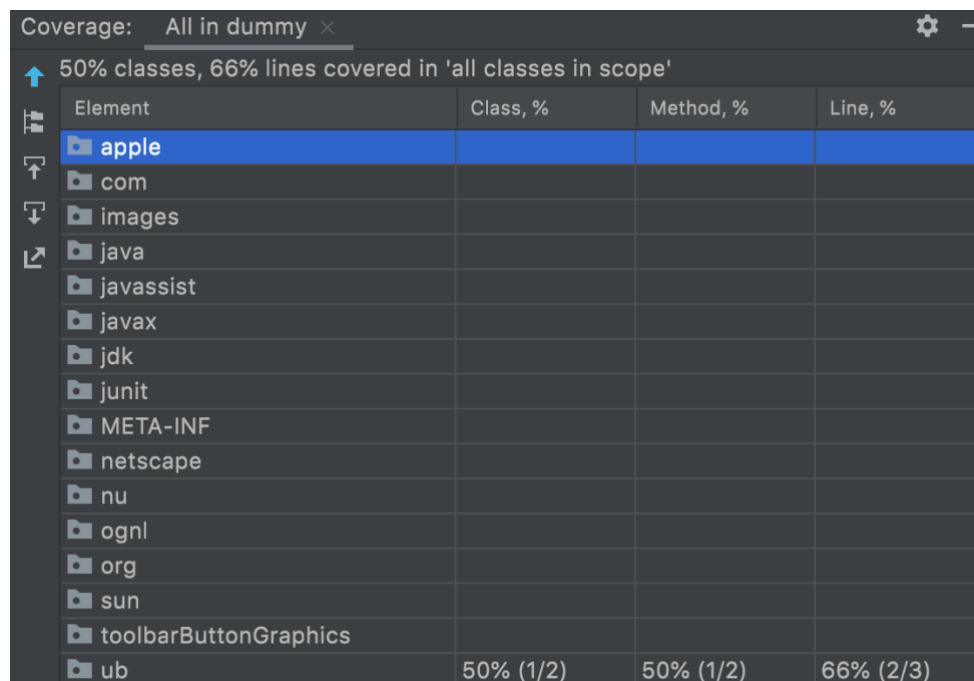
Description

Examples

Provides feature for adding two integers. For example 2 plus 5 should give 7.

Results generated by [Concordion](#)
in 89 ms on 16-Sep-2021 at 18:10:47 CEST

També es pot seleccionar "Run All Test with Coverage" del menú *Run* i dóna una taula amb el tant per cent del codi analitzat pels tests:



Coverage: All in dummy ×

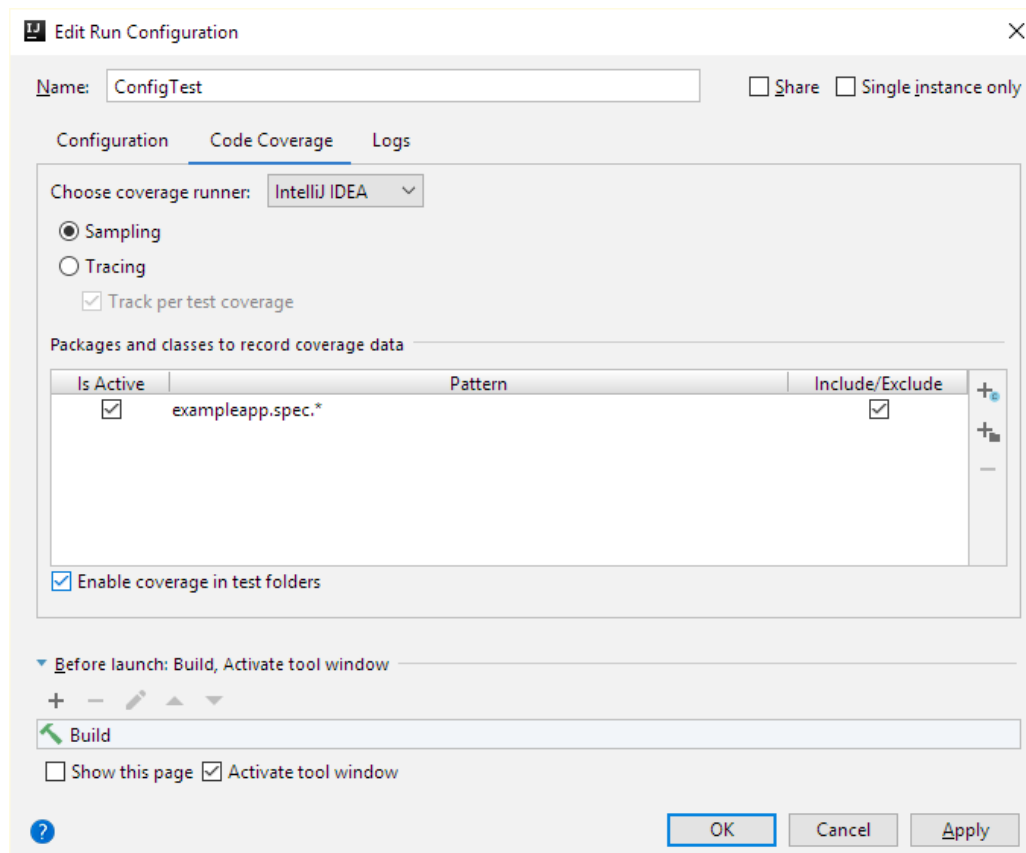
50% classes, 66% lines covered in 'all classes in scope'

Element	Class, %	Method, %	Line, %
apple			
com			
images			
java			
javassist			
javax			
jdk			
junit			
META-INF			
netscape			
nu			
ognl			
org			
sun			
toolbarButtonGraphics			
ub	50% (1/2)	50% (1/2)	66% (2/3)

Si al córrer el test *with Coverage* reveu el missatge següent caldrà una petita configuració addicional al projecte.

No coverage results. Click [Edit](#) to fix configuration settings.

Si feu clic al enllaç *Edit* se us obrirà un menú, i anant a la pestanya *Code Coverage* haureu d'habilitar la opció *Enable coverage in test folders*.



3. Referències i més documentació.

<http://concordion.org/tutorial/java/html/>

<http://www.methodsandtools.com/tools/concordion.php>

Disseny de Software

Curs 2021-22

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques i Informàtica. UB

<https://plus.google.com/+ConcordionOrganization>

<https://www.adictosaltrabajo.com/tutoriales/primeros-pasos-con-concordion/>