

Java & Spring Framework

[Inici](#) / [Els meus cursos](#) / [Java & Spring Framework](#) / [Sprint 1. Java Language](#) / [Tasca S1.04. Testing](#)

Tasca S1.04. Testing

Pendent: Rebre una qualificació



Nivell 1: JUnit

- Exercici 1

- Crea una classe Java que insereixi en una llista ordenada els noms dels mesos de l'any.
- Verifica amb JUnit que la llista té 12 posicions, que no és null i que en la seva posició 8 conté el nom 'agost'.

- Exercici 2

- Crea una classe anomenada CalculoDni que calculi la lletra del DNI en rebre el número com a paràmetre.
- Crea una classe JUnit que verifiqui el seu correcte funcionament, parametritzant-la perquè el test rebi un espectre de dades ampli i validi si el càlcul és correcte per a 10 números de DNI predefinits.

- Exercici 3

- Crea una classe amb un mètode que llanci una `ArrayIndexOutOfBoundsException`.
- Verifica el seu correcte funcionament amb un test JUnit.



Nivell 2: Hamcrest

- Exercici 1

Defineix un **Matcher** personalitzat per a Hamcrest que proporcioni el Matcher de longitud per a un String.

Volem usar la classe -> **FeatureMatcher**.

Amb FeatureMatcher podem ajustar un Matcher existent, decidir quin camp de l'Objecte prova ha de coincidir i proporcionar un missatge d'error agradable. El constructor de FeatureMatcher té els següents arguments en aquest ordre:

- El matcher que volem embolicar.
- Una descripció de la funció que provem.
- Una descripció del possible mismatch (desajustament).

L'únic mètode que hem de sobreesciure és `featureValueOf (T actual)`, que retorna el valor que es passarà al mètode `match ()` / `matchesSafely ()`. Utilitza el seu comparador personalitzat en una prova per a comprovar si la cadena "Mordor" té una longitud de 8.

Ajusta la prova si és necessari.



Nivell 3: AssertJ

Crea les classes Java i/o els tests que les proven per a demostrar el següent:

- Exercici 1

Una assertió que demostrï que el valor de dos objectes sencers és igual, i una altra que indiqui que no ho és.

- Exercici 2

Una assertió que demostris que la referència a un objecte és la mateixa que una altra, i una altra que indiqui que és diferent.

- Exercici 3

Una assertió que indiqui que dos arrays d'enters són idèntics.

- Exercici 4

Crea un `ArrayList` contenidor de diversos tipus d'objectes (crea'ls també). Escriu una assertió per a verificar l'ordre dels objectes en l'`ArrayList` segons han estat inserits.

- Ara verifica que la llista anterior conté els objectes en qualsevol ordre.
- Verifica que, en la llista anterior, un dels objectes s'ha afegit només una vegada. Deixa un dels elements sense afegir, i verifica que la llista no conté aquest últim.

- Exercici 5

Crea un map i verifica que conté una de les key que li afegeix.

- Exercici 6

Provoca una `ArrayIndexOutOfBoundsException` en una classe qualsevol. Crea una assertió que validi que l'excepció és llançada quan correspon.

- Exercici 7

Crea un objecte `Optional` buit. Escriu l'assertió correcta per a verificar que efectivament està buit.



Objectius

- Aprendre a manejar les llibreries d'assertions JUnit, Hamcrest i AssertJ.



Durada: 2 dies



Lliurament:

Envia la URL a un repositori Github que contingui la solució.



Estat de la tramesa

Estat de la tramesa

Cap intent

Estat de la qualificació

Sense qualificació

Darrera modificació

-

Entregar

Encara no heu fet cap tramesa.

