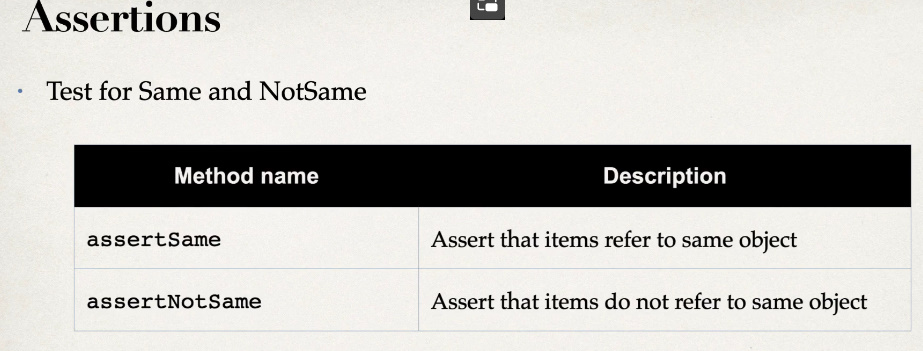
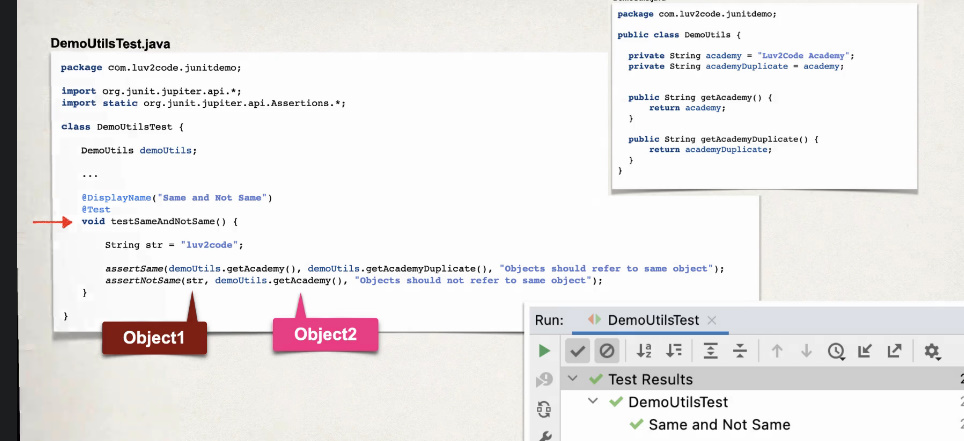
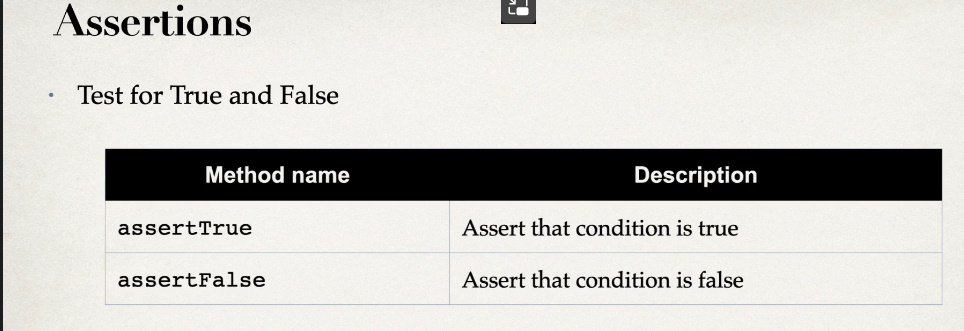
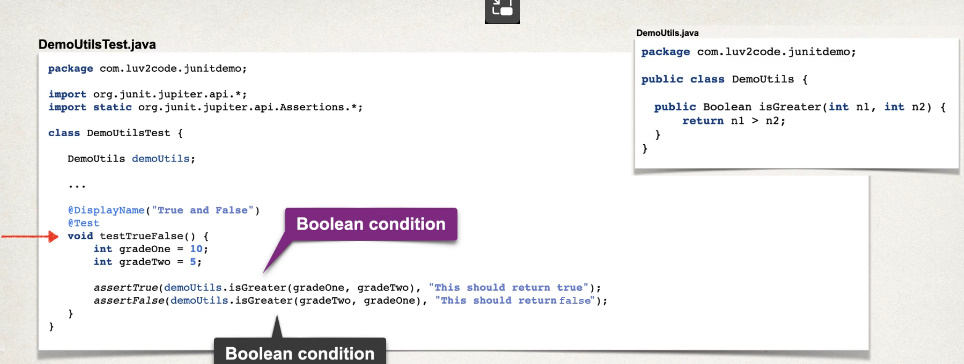
**assertSame si assertNotSame**

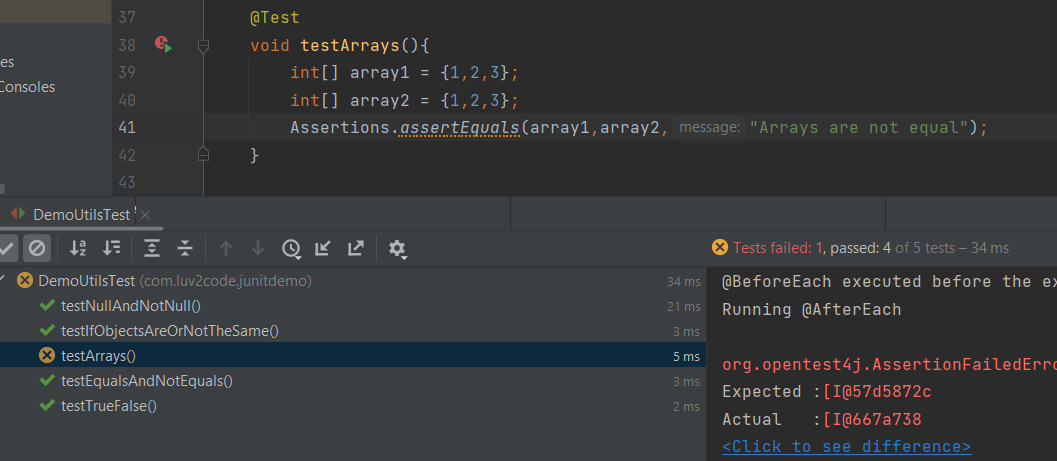
* 
* Deci, aceste metode pur si simplu verifica daca 2 obiecte sunt identice, adica daca variabilele care se compara au aceeasi referinta stocata catre obiect.
* assertSame/NotSame(objExpected, obj,message )
* 

**assertTrue si assertFalse**

* 
* 
* assertTrue/False(Expresie,mesaj);

**assertArrayEquals, assertIterableEquals and assertLinesMatch**

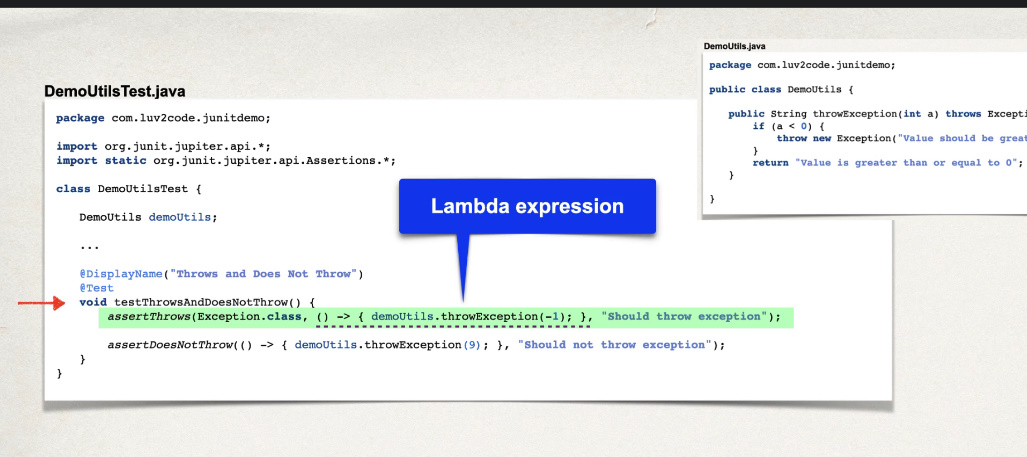
* assertEquals() poate lua si arrays, dar va compara iarasi referintele, de ex:



* Aici ne ajuta deja **assertArrayEquals(array1,array2,mesaj).** Acesta va compara fiecare valoare din arrays si daca toate sunt identice ca valoare, e bine, daca nu, testul estueaza. Pot fi si arrays de String si de orice. El nu compara referintele, ci valorile!
* **assertIterableEquals(iter1,iter2,mesaj)** – compara orice structuri de date daca au elemente identice, ca valori, nu ca referinte.
* **assertLinesMatch(iter1,iter2,mesaj)** – compara daca structurile de date au stringuri identice. **Deci e pentru a compaar stringuri doar!** Suporta si regulaexpressions si asta e ceva specific pentru el.

**assertThrows, assertDoesNotThrow and asserTimeout and assertTimeoutpreemptively**

* **assertThrows(ExceptionType.class,() -> {})** – se asigura ca sa arunca o exceptie de tipul dorit. Atentie! Al 2 parametru nu este o simpla metoda executata sau obiect, ci o lambda expression care face tot ce dorim noi ca sa se arunce expresia.



* **assertDoesNotThrow(() -> {}) –** se asigura ca lamba expression nu arunca nicio exceptie
* **assertTimeout(Duration.ofSeconds(timp),() -> {}, mesaj) -** se asigura ca o executarea unei actiuni sa nu depaseasca intervalul de timp stabilit cu Duration.ofSecond(timp).
* **assertTimeoutPreemptively(Duration.ofSeconds(timp),() -> {},mesaj) –** se asigura ca o executarea unei actiuni sa nu depaseasca intervalul de timp stabilit cu Duration.ofSecond(timp). Daca il depaseste, se opreste executia si testul da fail

**Atentie!** **assertTimeoutPreemptively** are o deosebire de **assertTimeout**. Deci, assertTimeout nu opreste fortat metoda din executie daca aceasta depaseste timpul maxim permis, ci va astepta pana ea termina executia, insa assertTimeoutPreemptively, dupa ce metoda depaseste timpul setat, va opri fortat executia ei, fara a astepta ca ea sa termine.

****

