**AssertJ**

**(assertek írása egyszerűbben(kollekciók), fluent API =egy utasítás, nem kell statikus import, sajátok írása vagy generálása)**

ha jUnitban **kollekciókra** vagy **javabeansekre** szeretnénk **ellenőrzéseket** végezni, akkor **korlátokba ütközhetünk**

az **Assertj** egy olyan **library**, amivel **egyszerűbben** tudunk ilyen **asserteket** írni

**Fluent API-val** rendelkezik -> **egy utasításba tudunk több állítást meghatározni** és a fejlesztőeszközök is segítenek, **nem kell statikusan importálni ezeket a metódusokat**, hanem ki tudja találni, hogy melyik assert metódusokat használhatunk

asssertj-vel lehet saját asserteket is írni vagy generálni

**Egyszerű assert**

**assertThat(), elvárt eredmény = nem második paraméter hanem további metódus hívás, egyszerre többet is; aktuális.elvárt vs elvárt, aktuális(sima assert)**

nem második paraméterként, hanem további metódus hívással adjuk meg az elvárt eredményt

assertEqualshoz képest először van az aktuális eredmény és nincs 2. paraméter

fluent API – egyszerre két assertet is megadhattunk egy utasításban!



**Kollekciók**

**(String lista és Employee lista -> extracting, stream és lambda, contains, containsOnly, isEqualTo)**



extracting() – employee listából csinál egy streamet és abból getname alapján String streamet

ugyanez lambdával is

**Tuple és szűrés**

**extracting több atrira egyszerre -> tuple(), filteredOn(), contains(), containsOnly()**

Több attribútum kinyerése; több érték -> **a tuple(=értéksor)** metódussal a containsen belül csak erre a két feltételre adunk meg értékeket

**.filteredOn()** -> szűrésre, lambdával



**Üzenetek testreszabása**

**(.as, ha elbukik, hozzá üzi placeholderrel, teljes üzi átírása: withFailMessage())**

**.as** metódussal meg lehet adni, hogy ha elbukik az összehasonlítás, mi kerüljön kiírásra; az assertJ általi üzenet előtt fog megjelenni, szögletes zárójelben

ha a teljes(assertj-s) üzenetet át akarjuk írni, akkor ezt a withFailMessage()-el

az **üzenetek paraméterezhetőek**, **megadhatunk placeholdereket** -> ’%s’; az utáni lévő többi paraméter értékével fogja a placeholdert kicserélni

****

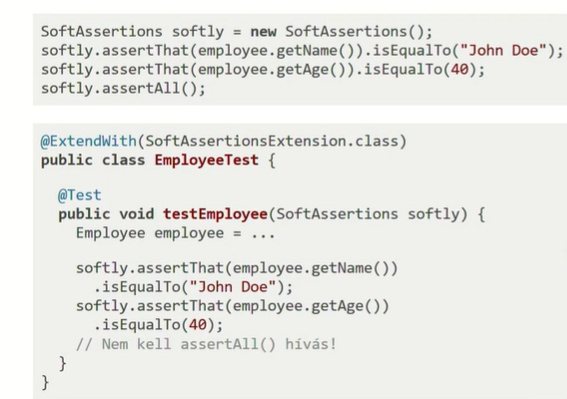
**Soft Assertion**

**(SotfAssertons példányra assertThat, assertAll() lezárás a végén; végigmegy és összesít, példányosítás nélkül: @ExtendWith és a softly átadása a teszt metódusnak)**

**nem bukik el azonnal** a teszteset, amikor egy feltétel nem teljesül, hanem ezeket **összegyűjti**, végigmegy az összesen és a **végén összesíti** az összes elbukott assertet

Ehhez példányosíytani kell egy Softassertions-t, ugyanolyan metódusai vannak, mint a sima assertnek,; meghívjuk az assertThat()-eket, majd az assertall()-nál fog leállni a végrehajtás és kiírásra kerülnek a bukottak

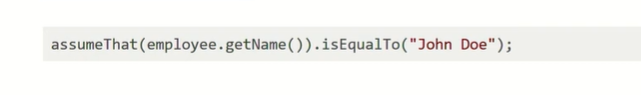
**lehet példányosítás nélkül is, @ExtendWith(SoftAssertionsExtension.class) és a teszt metódusnak paraméter átadás(=paraméter injection);** ebben az esetben nem kell assertall() hívás se a végén



**Assumption**

**(csak akkor fusson a teszt, ha valami feltétel teljesül)**

megáll a teszteset futása, ha a feltétel nem teljesül, de nem fogja hibásnak jelenteni az esetet; mikor használjuk? csak akkor van értelme a futásnak, ha valamiféle feltétel teljesül

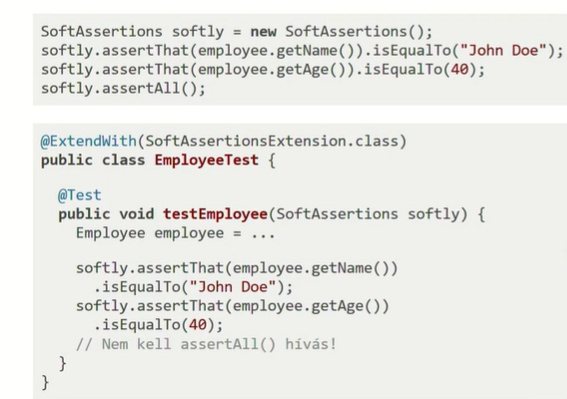


**pom.xml**

**(dependency felvétele test scope-al -> mit jelent)**

Ahhoz, hogy mindezt használni tudjuk, fel kell venni **az assertJ-t, mint dependencyt a pom.xmlbe test scopeal** -> kizárólag teszt esetek futtatásakor lesz benne az assertJ a classpathba, az alkalmazásba egyébként nem kerül bele





**Az AssertJ használata (gyakorlat)**

1. pom.xml függőség felvétele
2. tesztosztályban importálás assertJ
3. EmployeeServiceTest -> teestListemployees()

employes.get(0) -> assertThat..getname..isequalTo..

…startswith..endswith

1. .hasSize…extracting lamdával és contains, majd két paraméterrel és tuple, majd fusson hibára
2. .as az egész elejére, hogyx legyen hozzá szöveg, majd withfailmessage-el
3. majd két hibára futó assertthat a névre, elé softAssertions példány, és ezen meghívva és a végére assertAll(); ugyanezt annotációval és a teszt metódusnak paramétert átadva
4. soft megjegyzésbe, és AssumeThat import assertjből!, ne teljesüljön; nem fog hibára futni, csak nem futtatja le