Vad är Unit-Testing?

Ett test att köra på ett programs enskilda komponenter för att testa logiken. Förutsatt att programmets funktioner är uppdelade i enskilda moduler, dvs. att en metod bara gör en specifik sak. Detta för att kunna se så enskilda delar av programmet gör det dem ska. Istället för att köra hela programmet och hoppas att allting fungerar som det ska för att resultatet verkar stämma. Också för att man över tid ska kunna testa så enskilda delar fortsätter funka även efter att ändringar har gjorts på koden. Särskilt användbart när det kommer till komponenter som är beroende av andra komponenter för att fungera, komponenter som i vissa fall kan vara lokaliserade utanför programmet, så som en databas som man kopplar upp till. Istället för att vara beroende av en annan komponent så kan man göra ett simulerat fall att testa på. Man matar in ett förväntat resultat som metoden antingen lyckas eller misslyckas med att få fram, vid misslyckande får man det verkliga resultatet som kan ge ledtrådar om var felet ligger.

Annotation/Annotering

Annotations är särskilda nyckelord som används för att signalera till test-ramverket som används, i detta fall JUnit 5, att metoden som följer är en test-metod och att den ska kunna köras som ett test. Bra att tydligt döpa denna typen av metoder så det är uppenbart vad den används för, ex. testMethodForMultiplication eller shouldReturnTrue. När en annoterad metod körs skapar JUnit en ny instans av klassen och kör metoden I fråga. Om metoden ger några *exceptions* så misslyckas testet och detta skrivs ut av JUnit.

@Test

Är en sorts annotering som används mycket. Den indikerar, som nämnt, att metoden kan köras som ett test och tar emot två parametrar. Den första parametern är förväntat(expected) resultat. Det kan vara ett exception man förväntar sig...

@Test(**expected=IndexOutOfBoundsException.class**)

public void metodAttTesta() {

kod som ska köras

}

...Eller att man säger till JUnit att göra att testet misslyckas om det tar längre än en viss tid för metoden att köras...

@Test(**timeout=100**)

Assertion

En samling metoder I JUnit ramverket som används för att jämföra värde I/undersöka förväntat värde i objekt/primitiva typer. Den Information sparas från misslyckade assertions som visar hur resultatet avviker sig.

assertEquals

Används för att testa om två primitiva typer eller objekt är jämlika I värde/innehåll.