Assert

# Mi az assert?

Több helyen is találkozhatunk ezzel a kifejezéssel programozásban, különösen, ha unit tesztekkel van dolgunk. Ott az assert szerepe az, hogy megvizsgálja, hogy egy számított érték megegyezik-e valamilyen várt értékkel és jelezze a teszt, hogy sikeres-e avagy sem.

Java-ban azonban az assert-nek más szerepe is van, ebben a nyelvben ez egyben egy foglalt szó is. Arra jó, hogy a program hibát dobjon, ha valamilyen feltétel nem teljesül. Nyilván ez ekvivalens azzal, hogy írunk egy if-et, és ha nem igaz, akkor feldobunk egy kivételt. Az assert erre egy úgynevezett *shortcut-ot* (magyarul: rövidítést) szolgáltat, amellyel akár egy sorban is (természetesen ésszerű keretek között, olvashatóan) elvégezhetjük ezt a műveletet.

A használata a következő: fogjuk az assert kulcsszót és utána leírunk egy logikai kifejezést (feltételt). Ha a kifejezés hamissá értékelődik ki, akkor a programunk automatikusan egy AssertionError-t fog dobni, amely egy kivétel, azaz try-catch blokkal kezelhető. Ez további lehetőségeket tár fel a hibák kezelésére. Ahhoz, hogy az assert-et használni tudjuk, a programot az –enableassertions (vagy röviden csak –ea) paraméterrel kell futtatnunk, másképpen nem fog csinálni semmit.

# Példa

A példánkban egy olyan metódust szeretnénk írni, amellyel egy kört rajzolunk ki. A hogyan-t azt most nem tárgyaljuk, az assert használatán van a hangsúly.

Kód (Java):

public void drawCircle(double r, int x, int y) {

assert r >= 0 : "A sugár nem lehet negatív";

// rajzolás

}

Így, a metódus hívásakor, ha negatív sugarat adunk meg a körünknek, a program el fog szállni egy AssertionError-ral. Ezen kívül feltűnhet még a feltétel után szereplő szöveg, amely egy kettősponttal van tőle elválasztva. Lehetőségünk van üzeneteket feldobni a hibák mellett ezzel a szintaxissal. A példában, ha rossz paramétert kap a metódus, akkor feldobja a kivételt és hozzáteszi a következő üzenetet: „A sugár nem lehet negatív”.