

Java8: A Hash algoritmus változni fog

Egy hónapja került ki egy [levél a java7u-dev listára](#), amely szerint változni fog a *hash* képzés algoritmus a Java7u6 kiadással. Az új módszer a *hash* alapú tárolókat érinti, azokat is String típusú kulcsok esetén, mint például a HashMap, a Hashtable, a WeakHashMap, a LinkedHashMap, a ConcurrentHashMap, továbbá a Set implementációkat, mint a HashSet vagy a LinkedHashSet, illetve a járulékos osztályokat, mint a Properties.

Fontos tudni, hogy az alternatív *hash* algoritmus 512 elemszám felett lép működésbe (kivéve a ConcurrentHashMap, ahol csak ez az új algoritmus lesz), mivel a *hash* ütközések minimalizálása a cél és 512 elemszám alatt nem jellemző a *hash* ütközés. Alapvetően a *hash* algoritmus változása nem kellene befolyásolja egy szoftver működését, mivel ismert módon se a Map, se a Set, se a Properties nem garantálja sorrendet az iterációknál, ám a belső működés szerint a kulcsok alapján létrehozott *hash* értéke alapján került bejárásra a Map, így a sorrend valamilyen szinten garantált volt: és mivel garantált, így több helyen ki is használták ezt a programfejlesztők (nézzük át a saját fejlesztéseinket, hogy alapozunk-e sorrendiségre).

A lényeg: a Java7u6 kiadás megjelenésével figyeljük a potenciális hibákat, amelyek a fenti okokból következhetnek, ha megjelenik valami furcsaság, akkor a `jdk.map.altohashing.threshold` átállításával még paraméterezni tudjuk az alternatív hash algoritmus küszöbét, ám Java8 esetén elfordulhat, hogy nem lesz kikapcsolható, mivel küszöbérték függetlenül az új algoritmust fogja használni.