

# Tippek

Tippek, trükkök, praktikák a Java világából.



## Sonar metrikák

Auth Gábor posted on Jan 24, 2014

Aki használ Sonar <http://www.sonarsource.com/>-t (illetve mostanában már SonarQube <http://www.sonarqube.org> a neve, mert lett egy támogatott és egy közösségi kiadás belle), annak ismers lehet a magas számosságú Magic number [http://en.wikipedia.org/wiki/Magic\\_number\\_\(programming\)#Unnamed\\_numerical\\_constants](http://en.wikipedia.org/wiki/Magic_number_(programming)#Unnamed_numerical_constants) (jobban mondva Unnamed numerical constants) bejegyzés, mint Minor súlyosságú hiba. A Magic number – mint programozástechnikai hiba – elismert és létező jelenség,...

[2 Comments](#) ·

- [sonar](#)
- [metrics](#)
- [news](#)
- [tippek](#)



## Nexus nagytakarítás

Auth Gábor posted on Nov 16, 2013

Aki Sonatype Nexus <http://www.sonatype.org/nexus/>-t használ a build szerverén, mint Maven repository, azzal a bosszantó problémával szembesül egy idő után, hogy a sonatype-work könyvtár egyre több helyet foglal el. Ennek alapvetően két oka van: A saját termékünk kiadásait (releases) és munkapéldányait (snapshots) tartjuk itt a végtelenségig. A proxy repository helyi gyorsítóként funkcionálva letölti a távoli szerverekről az igényelt Maven artifact fájlokat,...

[7 Comments](#) ·

- [nexus](#)
- [maven](#)
- [tippek](#)
- [repository](#)
- [news](#)



## Getters/Setters - Use or Abuse?

Ketler István posted on Mar 27, 2013

There are many articles out there dealing with the accessor and mutator methods (getters and setters). They came together with the Object-Oriented principles. Information hiding says that the internal state of an object must be hidden from the outer world. That is, the fields it uses shouldn't be available. Yet, sometimes we have to query the state of the object, and even may want to change that state....

[1 Comment](#) ·

- [tippek](#)
- [getter](#)
- [setter](#)



## Reading data from XML file on mobile phones / A GWT for Mobile(mgwt) használatáról

Lehel Sipos posted on Oct 26, 2012

Röviden arról szeretnék beszélni, vagyis egész pontosan egy általam írt blogot prezentálni, melyben a "webkit"-et támogató mobil telefonokat tudjuk programozni a Java nyelv segítségével. Erre nyújt többek között lehetőséget, a "gwt for mobile(mgwt)" technológia is. Egy rövid példán/leíráson keresztül nézzük meg, mi is ez, hogy is működik ez a mostanában kifejlődő irányzat. <http://lehelsipos.blogspot.com/2012/10/mgwt-reading-data-from-xml-file-on.html> <http://lehelsipos.blogspot.com>....

- [tippek](#)
- [gwt](#)
- [mobil](#)



## GWT Animation - GWT FX / A Google Web Toolkit használatáról

Lehel Sipos posted on Oct 18, 2012

Google Web Toolkit (GWT) is a development toolkit for building and optimizing complex browser-based applications. <https://developers.google.com/web-toolkit/> <https://developers.google.com/web-toolkit/> You write your code in Java language and the GWT compiler will produce to you the client side javascript (/ajax) and html files. There are lot of additional java libraries which you can use to improve your GWT application for free: <http://code.google.com/p/gwt-google-apis/downloads/list> <http://code...>

- [tippek](#)
- [gwt](#)



#### [NIO.2 fájlmeletek](#)

[Auth Gábor](#) posted on Aug 12, 2012

A Java7 megjelenését megelőzően nagy várakozás övezte a NIO.2 újdonságait, aztán elmúlt a hype. Ennek oka valószínűleg az lehet, hogy a Java jelenleg két ers területén – az Android és a Java EE platformon – nincs igazán szükség a fájlmeletek támogatására. Az Android esetén megtrt dolognak számít a fájlrendszer, az SQLite alapú tárterület a támogatott; Java EE esetén pedig deklaráltan nem alapozunk a fájlrendszerre....

[1 Comment](#) ·

- [tippek](#)
- [java7](#)
- [nio2](#)



#### [OpenShift - JBoss AS 7.1 a felhben](#)

[Auth Gábor](#) posted on Jul 05, 2012

Egy ideje elérhet az OpenShift PaaS [http://en.wikipedia.org/wiki/Platform\\_as\\_a\\_service](http://en.wikipedia.org/wiki/Platform_as_a_service) lehetőség a RedHat oldalain, amelyet ingyen ki lehet próbálni: <https://openshift.redhat.com> <https://openshift.redhat.com> A regisztráció után egy pár lépésből álló varázslóval ki tudjuk választani a nekünk szükséges szolgáltatásokat: <image2012-7-5 21:11:43.png> Hozzuk létre a szolgáltatásunk domain nevét: <image2012-7-5 21:21:2...>

- [jboss](#)
- [red](#)
- [hat](#)
- [openshift](#)
- [cloud](#)
- [tippek](#)



#### [Automatikus Snapshot törlés Nexus-ból](#)

[Auth Gábor](#) posted on Jun 28, 2012

Ha használunk Sonatype Nexus <http://www.sonatype.org/nexus/-t> abból a célból, hogy a Maven alapú csomagjainkat biztos helyen tartsuk, akkor nem árt idnként ránézni a repository méretére, mert alapból a - SNAPSHOT csomagok nem törlődnek, így intenzív használat esetén a sonatype-work mérete az egekbe szökhet, mivel csak gyínek a feltöltött csomagok, amelyek közül általában csak az utolsót használjuk....

- [tippek](#)
- [nexus](#)
- [maven](#)



#### [Java7 - Fork és Join](#)

[Auth Gábor](#) posted on Jun 12, 2012

A Java 5 hozott néhány újdonságot a párhuzamos programozás terén, a Java 7 további könnyítéseket tartalmaz, amelyek közül az egyiket Fork/Join <http://docs.oracle.com/javase/tutorial/essential/concurrency/forkjoin.html> néven találjuk meg a dokumentációt böngészve, s a megoldás használatához csak három osztályt kell megismernünk: RecursiveAction <http://docs.oracle.com/javase/7/docs/api/java/util/concurrent/RecursiveAction.html> – visszatérési érték nélküli feladat RecursiveTask <http://docs.oracle...>

[4 Comments](#) ·

- [java7](#)
- [news](#)
- [tippek](#)
- [concurrency](#)



#### [Java7 - java.util.Objects](#)

[Auth Gábor](#) posted on Jun 08, 2012

A Java7 egyik csendes újítása a `java.util.Objects` <http://hg.openjdk.java.net/jdk7/build/jdk/file/9b8c96f96a0f/src/share/classes/java/util/Objects.java> osztály, amely közel tucatnyi statikus metódust ad az objektumok kezeléséhez, amely metódusok egy része nem esik zavarba, ha null értéket kap. Az `Objects` tipikus esete utility osztályoknak, final módosítóval van ellátva, illetve van egy privát konstruktora is: `private Objects() { throw new AssertionError("No java.util.Objects instances for you!..."); }`

[2 Comments](#) ·

- [news](#)
- [java7](#)
- [tippek](#)



### [JPQL vagy SQL lekérdezése Criteria példányból](#)

[Auth Gábor](#) posted on Jun 05, 2012

A JPA els kiadása nem tartalmazta a Hibernate felhasználói között népszerű Criteria lekérdezést, amely – kisebb változtatásokkal – része lett a JPA 2.0 specifikációnak. A `CriteriaQuery` <http://docs.oracle.com/javaee/6/api/javax/persistence/criteria/CriteriaQuery.html> nagyon jól használható olyan lekérdezésekhez, amelyeknél paraméterből jön a szűrni kívánt mező, vagy a rendezendő oszlop. Az egyetlen probléma a végrehajtott JPQL/SQL lekérdezése,...

[1 Comment](#) ·

- [tippek](#)
- [jpa](#)
- [jpql](#)



### [MD5 ellenrz összeg készítése](#)

[Auth Gábor](#) posted on May 22, 2012

Ritkán elforduló feladat egy fájl vagy egyéb stream tartalmából MD5 ellenrz összeget készíteni, ám viszonylag hamar meg lehet találni a jó megoldást (a try-catch blokkot mindenki képzelje a forráskód köré): `MessageDigest checksum = MessageDigest.getInstance("MD5"); byte[] buf = new byte[1024]; FileInputStream fis = new FileInputStream(file); while (true) { int len = fis.read(buf, 0, buf...`

[3 Comments](#) ·

- [tippek](#)
- [md5](#)



### [A switch, az enum és a null érték...](#)

[Auth Gábor](#) posted on Jun 19, 2011

Érdekes "hibába" futottam bele minap, mégpedig egy nem várt `NullPointerException` keletkezett egy switch utasításban, amely kissé váratlanul ért... Tekintsük meg az alábbi rövid ki programot: `public class SwitchEnumTest { public static void main(String[] args) { Enum testEnum = null; switch (testEnum) { case ONE: System.out.println("ONE"); break; case TWO: System.out.println("TWO"); break; default: System.out.println("other"); } } public enum Enum { ONE, TWO,...`

- [tippek](#)
- [enum](#)



### [A HashMap, a TreeMap és a Hashtable...](#)

[Auth Gábor](#) posted on Oct 21, 2009

...avagy a jó, a rossz és a csúf. A közös tulajdonságuk egyszer, implementálják a `Map` interfészt, s kulcs-érték mentén tárolnak adatokat, illetve a hatékony működést a kulcs hash értéke alapján végzik. Mi a különbség a `HashMap`, a `TreeMap` és a `Hashtable` között? A legelső implementáció a `Hashtable`, amely az 1.0 óta van jelen a Java nyelvben, mondhatni együtt ntt fel vele. Manapság eléggé mellzött szerepe van, szinte alig találkozunk a `Hashtable` alapon szervezett tárolókkal,...

[3 Comments](#) ·

- [tippek](#)
- [map](#)



### [EclipseLink Cache Coordination JMS segítségével](#)

[Auth Gábor](#) posted on Sep 21, 2009

A JPA implementációk (és eldjük - a Hibernate) nagyon sok felesleges munkát levettek az ember válláról, s az egyszerűbb ORM mveletek mellé idvel - az adatbázisokkal való munka során felmerült

problémák megoldásaként - egyre több eszköz került. Ilyen eszköz az Entity és/vagy Query Cache, amely a már beolvasott entitást adta vissza az adatbázisból való kérdés nélkül, illetve a már lefuttatott kérdésre az adatbázisból egyszer már elkért halmazt adta vissza ugyanilyen módon....

- [tippek](#)
- [jpa](#)



#### [Calendar.getInstance mítosz](#)

[Auth Gábor](#) posted on Jul 18, 2009

Fejlesztői beszélgetéseken sokszor elkerül, hogy idmérésre a Calendar használata kerülend, mivel túl sok erőforrást igényel a példányosítás, illetve a használata. A fejlesztők ebből az okból jobban szeretik a System osztály `currentTimeMillis` metódusát használni, amely egyszer UNIX időbélyeget ad vissza. A tesztelés egyszer, mérjük meg, hogy az egyes módszerek mennyi időt vesznek el a program életéből. A mérési keretrendszer egyszer,...

- [tippek](#)
- [mítosz](#)



#### [ThreadInfo lekérdezése](#)

[Auth Gábor](#) posted on May 23, 2009

A Java 6 megjelenésével több belső információhoz férhetünk hozzá futásidőben, amelyek közül érdekes lehet lekérdezni a szálak adatait, beleértve olyan metrikát, mint a felhasznált CPU idő. A `ThreadMXBean` a `ManagementFactory` statikus metódusai közül van, innen tudjuk lekérdezni a `ThreadMXBean` példányt, amelyből le tudjuk kérdezni az éppen létező szálak adatait: `ThreadMXBean threads = ManagementFactory.getThreadMXBean(); ThreadInfo[] threadInfos = threads.getThreadInfo(threads....`

- [tippek](#)



#### [OpenOffice integráció](#)

[Auth Gábor](#) posted on Feb 08, 2009

Aki már próbálkozott OpenOffice integrációval, annak hasznos lehet az `ODFToolkit` ismerete, mivel ehhez nem kell telepített OpenOffice a gépre, hanem az ODF szabvány ismeretében - némi XPath és XML tudással - képesek vagyunk olvasni és írni az OpenOffice által használt dokumentumokat. Az <http://odftoolkit.org> <http://odftoolkit.org> oldalon egész szerteágazó eszközkészlettel találkozunk, nézzük meg közelebbről az `ODFDOM` csomagot. Az `ODFToolkit` alapja az `ODFDOM`,...

[1 Comment](#) ·

- [tippek](#)
- [openoffice](#)



#### [Római számok](#)

[Auth Gábor](#) posted on Nov 25, 2008

Ritkán van szükség római számok használatára, legtöbbször soha nem is találkozunk azzal a helyzettel, amikor a feladat egy olyan programot írását követeli meg, amely az arab számainkat római számmá alakítja. Nézzünk meg egy elegáns megoldást. 

```
public class Main { public static final int[] NUMBERS = new int[] { 1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1 }; public static final String[] LETTERS = new String[] { "M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V", "IV", ...
```

- [tippek](#)
- [római](#)



#### [Java 6 - Queue és Stack](#)

[Auth Gábor](#) posted on Oct 26, 2008

A Java 6 nyelvi elemekben nem hozott akkora újrást, mint a Java 5, de jelentősen bővült a felhasználható eszközök száma, sok új osztály került a `java.util` csomagba, ilyen a `Deque` <http://java.sun.com/javase/6/docs/api/java/util/Deque.html> osztály is, amelynek a neve "Double Ended QUEUE" kifejezésből származik. Ez azt jelenti, hogy a `Deque` példányt használhatjuk `queue` és `stack` üzemmódban is. Csapjunk a közepébe, és nézzük az alábbi programot: 

```
import java.util.ArrayDeque; import java.util....
```

- [tippek](#)
- [queue](#)



### [ThreadInfo lekérdezése](#)

Unknown User ((k)risztián) posted on Oct 17, 2008

A Java 6 megjelenésével több bels információhoz férhetünk hozzá futásidben, amelyek közül érdekes lehet lekérdezni a szálak adatait, beleértve olyan metrikát, mint a felhasznált CPU id. A titok a ManagementFactory statikus metódusai körül van, innen tudjuk lekérdezni a ThreadMXBean példányt, amelyből le tudjuk kérdezni az éppen létező szálak adatait: ThreadMXBean threads = ManagementFactory.getThreadMXBean(); ThreadInfo[] threadInfos = threads.getThreadInfo(threads....

- [tippek](#)



### [Gyenge referencia](#)

Radics László posted on Oct 08, 2008

A java egy régi (1.2-es verziója óta létező) de kevésbé ismert részéhez tartozik a gyenge referencia. Igaz alkalmazási területe is elég szűk, de ha tudjuk, hogy léteznek és mire jók, a megfelelő időben elhúzva a zsebünkben nagyon hasznos tud lenni a java.lang.ref csomag. Ha egy mondatban össze kellene foglalni mi is az a gyenge referencia akkor úgy írnám le, hogy olyan hivatkozás egy objektumra, ami nem akadályozza meg a szemétdijyt (garbage collector),...

- [tippek](#)
- [weakreference](#)



### [A synchronized mítosz](#)

Auth Gábor posted on Jul 29, 2008

Java programozók körében elterjedt tévhit, hogy a synchronized módosítóval ellátott metódusok vagy blokkok által bezárt programsorokat csak egy szál futtathatja egy időben. A valóság az, hogy a kérdéses osztály példányaira vonatkozik a synchronized, ha az osztályból van két példányunk, amelyek egy közös statikus változót használnának, akkor mit sem ér a synchronized... Hogy ne a levegőbe beszéljek, nézzünk egy konkrét példát, vegyünk például egy osztályt,...

[4 Comments](#) ·

- [synchronized](#)
- [mítosz](#)
- [tippek](#)



### [Hello World program - main metódus nélkül](#)

Auth Gábor posted on Jul 26, 2008

Minden programozási nyelven vannak olyan trükkök, amelyek a nyelv határait feszegetik, vagyis teljesen legális nyelvi eszközökkel érnek el meglepő viselkedést. Ilyen lehet egy olyan rövid program, amely kiírja a klasszikus "Hello, World!" szöveget, de nincs benne main metódus, ahol ezt megtennénk. Mégis lehetséges... A megoldás egyszer, a ritkán használt statikus inicializátor blokkba kell tennünk a "Hello, World!" kiírást. Ez a blokk akkor hajtódik végre egyszer,...

- [trükk](#)
- [tippek](#)



### [Polimorfizmus mítoszok](#)

Auth Gábor posted on Mar 06, 2008

Az objektum orientált programozás terjedésével a monolitikus programokban hívk egyik nagy ellenérve az örökléssel és a polimorfizmussal kapcsolatban a túlzott erőforrásigény volt. A JavaSpecialist <http://www.javaspecialists.eu/archive/Issue157.html> blog gazdája készített egy mérést, amely alapján mi is készítettünk méréseket, amelyek meglepő eredménnyel zárultak. Egészséges Java öntudattal az ember azt kell gondolja,...

- [tippek](#)
- [mítosz](#)
- [polimorfizmus](#)