# Vegyük el a Java étvágyát!

Thomas Einwaller elmélkedett a hatos Java újdonságairól, amelyekkel meg tudjuk találni a Java alkalmazásunk memóriazabáló részeit. Persze most *minde nki* felhördül, hogy a Java hogy lehet memóriazabáló... amikor ott az automatikus szemétgyjt. Ha nem tördünk az példányainkkal megfelel gondossággal, akkor elfordulhat, hogy a memóriában maradnak, mert van még rájuk referencia (akár körkörös is lehet!). Ezernyi módja van a Java programunk étvágyának... és alig van néhány módszerünk a hiba megkeresésére. Els lépésként futtatnunk kell a **jmap** parancsot, amelynek megadjuk a futó JVM PID számát, illetve pár egyéb paramétert:

```
[javaforum@javaforum:~]$ /opt/SDK/jdk/bin/jmap -dump:live,format=b,file=javaforum.bin 13878
Dumping heap to javaforum.bin ...
Heap dump file created
[javaforum@javaforum:~]$ ls -l javaforum.bin
-rw------- 1 javaforum other 106338209 Aug  7 20:47 javaforum.bin
```

A program futása során egyszeren készít egy dump-ot a JVM heap területérl, amelyben aztán a **jhat** program kedvére turkálhat:

```
[javaforum@javaforum:~]$ /opt/SDK/jdk/bin/jhat -J-mx768m javaforum.bin
Reading from javaforum.bin...
Dump file created Tue Aug 07 20:47:51 CEST 2007
Snapshot read, resolving...
Resolving 891772 objects...
Chasing references, expect 178 dots... [...]  ....
Eliminating duplicate references... [...] ...
Snapshot resolved.
Started HTTP server on port 7000
Server is ready.
```

A gépen, ahol futtatjuk a **jhat** programot, kinyílik egy 7000-es számú port, amelyen egy egyszer webszervert futtat a **jhat**, ahol végig tudjuk kattintgatni a JVM lementett állapotát, érdemes megnézni, hogy melyik osztályból mennyi példány létezik, és azokra honnan kerül hivatkozás... és még sok egyéb információt is ki tudunk bányászni az adatrengetegbl.