

Egyszerű divide and conquer szálakkal

1. Futtasd az előadás diasorának programjait.
 1. Figyeld meg, mi történik, amikor két szálon különböző szövegeket íratsz ki végtelen ciklusban (pl. "Hi!" és "Bye").
 - Hogy le is álljon a program, “végtelen” helyett elég lehet pl. százezer iteráció.
 2. Figyeld meg mi történik, ha hosszú szövegeket ír ki a program (pl. 200 db '*' és 200 db '_' karakterből álló szövegeket).
 3. Mi történik, ha **Thread.yield()** hívást teszel a szál kódjába?
2. Írd át a fönti programokat ekvivalens módon, csekély mértékben eltérő stílusban.
 1. Úgy, hogy **Runnable** implementációt adsz át a szál konstruktorának.
 2. Úgy, hogy **lambda-kifejezést** adsz át a szál konstruktorának.
3. Készítsd el a prímszámkereső algoritmus párhuzamosított változatát. Módosítsd a korábban elkészített naiv algoritmust a következőképpen:
 1. Indíts egy szálát, ami megkeresi a prímszámokat **2** és **$n/2$** között, miközben a fő szál keres **$n/2$** és **n** között. A fő szál várja be a másikat (**Thread.join()** hívással), és fűzze össze az eredményeket.
 2. Módosítsd a prímszámkereső programot azzal, hogy elindítasz annyi szálát, ahány processzormaggal rendelkezik a géped, és oszd szét a keresési tartományt ezen szálak között.
 - Tipp: használd a **Runtime.getRuntime().availableProcessors()** hívást.