

A gyakorlatokon az alábbi feladatokból fogjuk megoldani néhányat, illetve a kurzus csatornáján is szerepelhetnek még további feladatok. A közösen nem tárgyalt feladatokat önálló gyakorlással lehet megoldani, és lehet kérdezni velük kapcsolatban a gyakorlatokon.

Egyszerű szálkezelő feladatok

1. Készítsünk egy programot, amelyben a `main()` metódus dob egy `RuntimeException`-t.
 - 1.1. Vizsgáljuk meg a hibaüzenetet, figyelmet fordítva a "main thread" kifejezésre.
2. Készítsünk egy programot, amelyben a `main()` metódus elindít egy szálat, ami dob egy `RuntimeException`-t.
 - 2.1. Látunk-e hibaüzenetet?
 - 2.2. Kapjuk el a kivételt a `run()` metódusban és hívjuk meg a `printStackTrace()` függvényt. Értelmezzük a hibaüzenetet.
3. Készítsünk egy programot, ami `n` szálat indít el, amelyek nem csinálnak semmit sem.
 - 3.1. Mekkora lehet `n` maximális értéke a futtató gépen?
4. Készítsünk egy programot, ami elindít `n` szálat, amelyek mindegyike elindít egy végtelen ciklust.
 - 4.1. Ez a program veszélyes lehet: kezelhetetlenné válhat a számítógép, újraindítás lehet szükséges.
 - 4.2. Most mekkora lehet `n`?
5. Készítsünk egy programot, amely elindít egy szálat és terminál. Az elindított szál is indítson egy másik szálat, majd termináljon, majd így tovább. Mi történik?
 - 5.1. Készítsünk egy változatot a programból, amely hasonlóan indít szálakat, de véges.
 - Tipp: adjunk hozzá egy konstruktorparamétert a `thread` osztályunkhoz, ami meghatározza, hogy milyen mélységig folytatódjon a lánc.
6. Készítsünk egy "fork-bomb" programot; elindít egy szálat, ami aztán elindít még két szálat, amelyek szintén elindítanak két szálat és így tovább...
 - 6.1. Ez a program veszélyes lehet: kezelhetetlenné válhat a számítógép, újraindítás lehet szükséges.

(a további feladatok még készülnek)