**Kedves Jelentkező!**

Az ITself Europe Informatikai Kft. egyik teszt feladatát tartod a kezedben, amely segítségével szeretnénk kiválasztani leendő munkatársainkat. Kérjük az anyag részletes átolvasását!Esetleges kérdéseidet a **support@itself.hu** címen teheted fel. A lentiekben leírt feladat reményeink szerint segít a szakmai alkalmasságod és rátermettséged objektív megítélésben.

# A Feladat leírása

A feladatot java nyelven kell megoldani. Egy olyan statikus interfészekkel rendelkező osztályt kell elkészíteni amely *n* darab erőforrás kiosztásának ütemezését oldja meg *k* darab kliens részére **kétféle műveletre** (írás/olvasás) **holtpont-, és éheztetésmentesen**. Mind az *n* erőforrásból 1-1 darab van. Ha egy erőforrás írásra van lefoglalva, akkor azt más kliensek semmilyen célra nem kaphatják meg. Amennyiben egy erőforrás olvasásra van lefoglalva, akkor azt olvasók megkaphatják, de írók nem. Az erőforrások a lefoglaláskor kapott kulccsal tehetők megint szabaddá.

Az erőforrások hivatkozásait *String* -ekkel reprezentáljuk (nevek).

A kliensek felé az erőforráskezelőnek két interfésze van a *getLock()*, és a *releaseLock()*. A *getLock()* interfész kötelező paramétere egy objektum (*Set<String>)*, melyben erőforrások nevei vannak és a kívánt művelet (*boolean: true* - írás*, false* - olvasás), visszatérési értéke egy azonosító (*long*).Amíg az erőforráskezelő nem tudja odaadni a kért erőforrásokat addig az adott klienst blokkolnia kell a *getLock()* híváson. Akkor térhet vissza, amikor a kért erőforrások elérhetőek a megadott művelethez.A *releaseLock()* interfész visszatérési értéke *void*, kötelező paramétere pedig egy azonosító, amit a kliens a *getLock()* hívásakor kapott.   
A kliensek az erőforrások foglalását/feloldását az *n* erőforrás részhalmazaiként kérik az erőforráskezelő osztálytól (a *getLock()* interfészen keresztül), és az itt kapott azonosító alapján engedik el (a *releaseLock()* interfészen keresztül).

A kliensek működése a következő:

* létrehozáskor kapnak egy műveletet (írás/olvasás) és egy véletlen elemszámú részhalmazt az *n* erőforrásnévből,
* meghívják ezzel a részhalmazzal és művelettel az erőforráskezelő *getLock()* függvényét. Amikor a *getLock()* függvény visszatér, a kliens várakozik 1000-2000ms között egy random értéket,
* az előbb lefoglalt erőforrásokat *releaseLock()*-al felszabadítja és ismét várakozik 1000-2000ms közötti randomot majd újraindul.

Az erőforráskezelőnek a következő feltételeknek kell megfelelni: éheztetésmentes, holtpontmentes, „demokratikus”.   
Az ütemezéskor a kéréseknek **alapvetően időrend alapján** kell következniük, azzal a feltétellel, hogy az algoritmus éheztetés-, és holtpontmentes kell legyen! A feladatban csak az erőforráskezelőt kell megírni melyhez a kliens osztály (*itself.probafeladat14.Client)* és indító függvénye (*main)* adottak.

A bemutatóprogram működése a következő:

* Inicializálni az erőforráskezelőt (ez legyen automatikus),
* A klienseket létrehozó osztály létrehoz *k* darab klienst véletlenszerű művelettel és erőforrásigénnyel, majd elindítja őket.

# Feladat bemutatása

A feladat bemutatására (a működési és programozástechnikai megoldások ismertetése) a közösen egyeztettet időpontban és helyszínen, kerül sor!

A bemutató előtt a jelentkező lehetőséget kap az általa szükségesnek ítélt telepítési, üzembehelyezési és tesztelési folyamatok elvégzésére.

A bemutató értékelési szempontjai sorrendben a következők:

* a feladat-kiírásnak megfelelő működés,
* logikus felépítés,
* stilisztikai alkalmazkodás a meglévő forráshoz,
* kreatív, hatékonyságot növelő, ötletes programozói megoldások,
* gyors, hiba mentes futás,
* előadói stílus.

Megjegyzés: A program üzembehelyezésére fordított idő tervezésekor figyelembe kell venni, hogy az előre egyeztetett időpont a bemutató kezdetére vonatkozik.

Sikeres munkát kívánva!

ITself Europe Informatikai Kft.