

1. Feladat – Lock (40%)

Készíts egy osztályt, ami lehetőséget ad erőforrás lefoglalására. Az osztálynak két metódusa van a *lock()* és az *unlock()*.

Ha egy szál meghívja a *lock()*-ot az akkor tér vissza ha megszerezte a futás jogát (azaz másik szál még nem szerezte azt meg előtte, amennyiben már egy másik szál lefoglalta a *lock*-ot akkor a szál várakozik míg az fel nem szabadul), a *lock()* azonnal visszatér ha a meghívó szál már birtokolja a *lock*-ot (ennek eldöntésében segíthet a *Thread.currentThread()* függvény).

Ha egy szál meghívja az *unlock()* függvényt, akkor ha a meghívó szál birtokolja a *lock*-ot a szál felszabadítja azt és erről értesíti a várakozó szálakat. Ha nem a *lock*-ot birtokló szál hívja meg az *unlock()* metódust, akkor *IllegalMonitorStateException*-t dob.

Teszteléshez indíts 5 szálát, amelyek mind 1 *lock* megszerzéséért versengenek. Amíg birtokolják a *lock*-ot 5 másodpercet *sleep*elnek majd közlik, hogy a “<sorszám>. szál végzett”, ezután elengedik a *lock*-ot.

Ahhoz, hogy az implementációd pozitív pontszámot érjen, egyáltalán ne használd a *java.util.concurrent* csomag osztályait! A feladat megoldása során figyelj oda, hogy a programod szálbiztos legyen, valamint ne használj se busy waitinget se pollingot. Ügyelj arra, hogy ne alakulhasson ki deadlock vagy livelock valamint - a lehetőségekhez mérten - valódi párhuzamosság legyen benne.

2. Feladat – IP Chat (60%)

Ebben a feladatban egy egyszerű IP Chat programot kell megvalósítanod. Mivel ZH üzemmódban nem tudod használni az internetet, így IP cím helyett localhost-on, portszámra fogsz tárcsázni. A programhoz szükséges grafikus felhasználói felületet is készíteni.

Induláskor generáljon egy random portszámot 16000 és 17000 között, ahol a bejövő hívásokra figyel és ezt jelenítse meg a felhasználói felületen is. Mellette legyen egy szövegmező, ahova egy másik program portszámát beírva és a hívás gombot lenyomva hívást kezdeményezhet. Amennyiben jó a hívott port, akkor építse fel a kapcsolatot, ha nem sikerült a kapcsolódás, akkor egy hibaüzenetet jelenítsen meg a felhasználó számára, ami után új hívással próbálkozhat. Ha hívás érkezik a programba, akkor álljon le a hívások fogadása és építse fel a kapcsolatot a partnerrel.

A felhasználói felület tartalmazzon egy szövegmezőt, ahova a chat üzeneteket tudja írni a felhasználó, valamint egy küldés gombot, amire kattintva elküldi azokat. Az elküldött és a beérkező üzenetek is jelenjenek meg valahogy megkülönböztetve. (Pl. írja eléjük, hogy “Partner:”, “Te:”).

Próbáld ki a programot! Indíts két programot és hívd fel az egyikkel a másikat, majd chatelj! A javítás során mi is teszteljük, kipróbáljuk a programot, így a maximális pontszám eléréséhez fontos, hogy a tanult technikák használatával és stabilan működjön az alkalmazás.