



Szolgáltatás Orientált Programozás gy.

Zárthelyi dolgozat 2024.

november xx.

A csoport

IP címek – Blockingcollection és ThreadPool

BlockingCollection és ThreadPool használatával oldja meg a következő termelő-fogyasztó problémát! A termelők IP címeket állítanak elő. Négy típusú termelő van: A, B C és D, amelyek csak ilyen osztályú címeket állítanak elő. Mindegyik **különböző mennyiségben** állítja elő ezeket. Az A osztályú cím előállításához kell a legkevesebb idő, a legtöbb a D osztályú címhez (Thread.Sleep). A fogyasztóknak is négy típusa van, A, B, C és D, akik csak ilyen osztályú címeket vesznek ki a pufferből. A berakásról és a kivételről is üzeneteket írnak ki, minden címtípushoz különböző szín tartozzon. A fogyasztók valamennyi címet feldolgozzák.

A kollekciókhoz (4 darab van) a hozzáférést, a termelők és a fogyasztók kontrollálását a Supervisor osztály végezze. Ebből egy van!

A termelő osztály mezői:

IP_type. értéke A vagy B vagy C vagy D – Char típus. Mit állít elő.

Colour: 4 szín lehet, minden típushoz más. ConsoleColor típusú

Amount: hány címet kell előállítania.

WorkTime: a „termelés” után hány millisec-et kell várni. (Lehet előtte is.: -). int.

A fogyasztó osztály mezői

IP_type: értéke A vagy B vagy C vagy D. Char típus. Mit fogyaszt.

Colour: 4 szín lehet, minden típushoz más. ConsoleColor típusú

IP Address classes:

- A. 1.0.0.0 – 126.255.255.255.
- B. 128.0.0.0 – 191.255.255.255
- C. 192.0.0.0. – 223.255.255.255
- D. 224.0.0.0 – 239.255.255.255

Ezeket a Random osztály segítségével generálja. Négy termelő és négy fogyasztó legyen! (És persze négy tároló, különböző kapacitásokkal!) A fenti korlátokon kívül más megkötés nincs.

Három és négy (ThreadPool, Task és dedikált szál)

Deklaráljon egy N elemű vektort, tölts fel a vektort (1-N közötti) véletlenszámokkal a Main-ben! (100-9999 közé eső számokkal.) Két szál segítségével számolja meg, hogy a vektorban hány hárommal osztható, de néggyel nem osztható, 4 jegyű szám van. Ehhez egy változót használjon, mindkét szál ezt a változót használja. Az egyik a vektor elejéről, a másik a vektor végétől kezdje el a keresést. Nem használhat lock-ot vagy Monitor-t! A végén a Main írja ki, hogy a megadott számokból mennyi volt a vektorban! A feladatot 3 eszközzel oldja meg: ThreadPool, Task és dedikált szálak. A Stopwatch segítségével mérje meg, és írassa ki, hogy melyik eszközzel milyen gyorsak az implementált algoritmusok!

ChatGPT4

Készítsen egy olyan programot, amelyben 2 metódus külön szálon futva egy listához ad elemeket, illetve vesz ki belőle.

Az első metódus 100.000 darab random számot ad a listához, ezek a számok pozitív egész számok, maximum 4 jegyűek!

A második szál megkeresi (0. indextől kezdve) az első legalább 3 jegyű számot, és helyére a listába -1-et ír. Ha nem talál ilyet,

akkor a szál azonnal álljon le!

A főprogram futtassa le a két metódust "párhuzamosan" és írja ki, hogy hány szám maradt a listában, és ezek közül hány darab az egyjegyű szám!

A megoldáshoz nem dedikált szálakat (ThreadPool) használjon! A lenti megoldást javítsa ki a specifikáció alapján! A felesleges objektumokat törölje!

aries.ektf.hu/~ksanyi/SOP/2024/ChatGPT4.zip (letölthető)

ZIP fájl, neve a neptunkod.

<https://oc.uni-eszterhazy.hu/s/GYQHnSXISPO0uUR>