Igaz-hamis

1) .Net környezet folyamat kilépésekor bevárja az összes előtér szálat.

IGAZ

2) Amikor egy AutoResetEvent objektum jelzett lesz, a rá várakozó szálak közül csak 1 futhat tovább.

IGAZ

3) Egy 32 bites számmal történő egyszerű értékadás .NET környezetben mindig atomi.

IGAZ?

4) .NET környezetben 2 folyamat között kölcsönös kizárás a lock utasítással megvalósítható.

IGAZ

5) Ha adott két objektum és ezekre a lock utasítással eltérő sorrendben szerez zárat két szál, nem

alakulhat ki holtpont.

HAMIS

6) .NET több nyelvet is támogat.

IAGZ

7) .NET NuGet csomagok formájában lehet kiegészítő könyvtárakat letölteni.

IGAZ

8) .NET Framework alapú konzol alkalmazások Linuxon is futnak

HAMIS

9) .JIT köztes kódot fordítja natív gépi kódra.

IAGZ

10) A közös dll-ek egy központi mappába történő telepítésének egyik előnye, hogy így minden esetben elkerülhető dll kell probléma.

HAMIS

Kisfeladatok

1) Mit értünk horganyzás (anchor) fogalmán Windows Forms alkalmazások esetén?

Az adott elemet (pl Button) egy adott távolságra állítjuk be a tartalmazó panel széltől/széleitől.

2) Adja meg egy mondatban, Windows Forms környezetben milyen megközelítést érdemes választani egy olyan saját vezérlő készítéséhez, mely maga is több vezérlőt tartalmaz!

UserControl megoldás leszármazás

3) Egy Windows Forms alapú alkalmazásban a célunk egy összetett (több vezérlőből álló), újrafelhasználható vezérlő kialakítása. Mit használna ehhez?

UserControl megoldás leszármazás

Nagyfeladatok

1) News Server hírszervert reprezentál. Egy új hírt a PublicNews, a művelet kivételekkel jelzi, ha valami null. Adja meg a NewsServer osztály teljes C# forráskódját, amely létrehoz egy példányt, feliratkozik az eseményre, majd kiírja az esemény paramétereit a konzolra.

2) Az (50,50) koordinátában zöld színű 1 pixel vastag folytonos vonallal rajzolt, kitöltetlen, 100 pixel oldalhosszúságú négyzet megjelenítése. “X” billentyű lenyomására a négyzet vonalának színe 10 másodperc alatt fokozatosan menjen át kékbe! Csak időzítő (Timer) alapú megoldás elfogadható!

3) Készítsen az alábbiaknak megfelelő C# nyelvű konzol alkalmazást!

Tegye fel, hogy egy már megírt kód szálakat indít (vagyis a szálak indításával nem kell foglalkoznia). Mutasson elegáns, hatékonyan működő C# nyelvű Console – vagyis mem Windows Forms – alapú megoldást a következőre: Egy folyamatosan futó termelő szál egytől kezdve (kb.) másodpercenként egy eggyel növekvő egész számot ír egy gyűjteménybe. A folyamatosan futó és várakozó fogyasztó szálak közül - ezekből többet is indíthat a környezet, akár egymással párhuzamosan - a legelső, mely értesül arról, hogy a gyűjtemény elemszáma elérte a 30-at, mielőbb írja ki a " megtelt" szöveget a konzolra, törölje az elemeket a gyűjteményből, majd lépjen ki. Az egyes szálak indulási sorrendjével kapcsolatban nem tehet fel semmit! Írjon egy olyan függvényt is, mellyel a rendszerben futó bármelyik szál bármikor lekérdezheti a gyűjtemény aktuális elemszámát!

Írja meg a termelő és fogyasztó szál kódját, valamint egyéb szükséges változókat, inicializálásokat - a szálak indításának kivételével. A megvalósítás során felhasználhatja a .NET keretrendszer List<T> osztályát (Add, Count és Clear műveletek)