Szoftvertechnikák BSc vizsgadolgozat

2013.06.05

Jegy:

Ponthatárok	
0-44	Elégtelen (1)
45-55	Elégséges (2)
56-70	Közepes (3)
71-84	Jó (4)
85-	Jeles (5)

Pontok					
kérdés	max	pont	kérdés	max	pont
1	11		6	13	
2	12		7	12	
3	12		8	14	
4	14		Σ	100	
5	12				

1) Felügyelt környezetek
   a) Jellemezze az azonosított (erős névvel ellátott) .NET szerelvényeket! (7p)
   b) Adja meg röviden, mit jelent a szerelvények vonatkozásában a .NET alkalmazások integritásvédelme, és hogyan kerül ez megvalósításra! (4p)

2) A DataHandler osztálynak van egy DoOperation nevű függvénye, mely paraméterként két egész számot és egy delegate-et kap. A DoOperation függvény összeszorozza a két számot, majd a paramétereket és az eredményt a paraméterként kapott delegate meghivásával naplózza (a naplózás elvégzése a delegate feladata).
   Adja meg a DataHandler osztály teljes kódját, illetve adjon példát a használatára: a példaprogram hívja meg a DoOperation függvényt egy olyan függvény megadásával, mely a paramétereket és az eredményt a konzolra naplózza! (12p)

3) Szálkezelés
   a) Adja meg a szálbiztos (thread-safe) osztály fogalmát egy-két mondatban! (2p)
   b) Adott az alábbi kódrészlet:

```
class Processor {
        List<int> numbers = new List<int>();
        static void Main() { ... } // belépési pont
        ...
}
```

   Egészítse ki a következőknek megfelelően: a főszál indítson egy munkaszálat, tegyen be egy tetszőleges számot az numbers listába, ezt jelezze a munkaszálnak, majd várja meg, míg a munkaszál befejezi a futását. A munkaszál az indulása után hatékonyan várakozzon a főszál jelzésére, a jelzés után vegye ki a listából az elemet, írja ki a konzolra, majd fejezze be a futását. A megvalósítás során ne feledkezzen meg a kölcsönös kizárás biztosításáról sem, amennyiben szükség van rá! (8p)
   c) Adja meg egy mondatban, mi történik egy szállal, ha a szál objektumára meghívjuk az Interrupt() műveletet! (2p)

4) Eseményvezérelt programozás és grafikus megjelenítés.
   a) Milyen következményei vannak annak, ha két űrlap birtokos/birtokolt (owner-owned) viszonyban van egymással? (3p)
   b) Írjon olyan C# nyelvű alkalmazásrészletet, amely az (50,50) koordinátában megjelenít egy zöld színű 1 pixel vastag folytonos vonallal rajzolt, kitöltetlen, 100 pixel oldalhosszúságú négyzetet. Ha a felhasználó lenyomja az „x" billentyűt, a négyzet vonalának színe 10 másodperc alatt fokozatosan menjen át kékbe! Csak időzítő (Timer) alapú megoldás elfogadható! (11p)

5) Adott az alábbi modell: Egy ALKATRÉSZT egy TERMÉKBE építenek bele, egy TERMÉKBE több ALKATRÉSZT. Egy ALKATRÉSZNEK több BESZÁLLÍTÓJA van, egy BESZÁLLÍTÓ több ALKATRÉSZT is beszállít. Az ALKATRÉSZNEK van neve (nem feltétlen egyedi!) és leírása, a BESZÁLLÍTÓNAK címe, a TERMÉKNEK sorozatszáma (egyedi). Adjon meg UML diagramot a modellhez, valamint képezze le adatbázis táblákba. A megoldáshoz fűzzön rövid szöveges magyarázatot, melynek során pontosan adja meg, hol használ idegen kulcsokat, és azok mire hivatkoznak! (12p)

6) Tervezési minták
   a) Adja meg röviden, hogy miben és hogyan segítenek a **tervezési** minták a tervezés során (de ne a tervezési minta definícióját adja meg) ! (3p)
   b) Jellemezze a "Memento" tervezési mintát! Mire ad megoldást a "Memento" tervezési minta? Mutassa be konkrétan **vagy** egy példán keresztül a minta működését! Ezen belül rajzolja fel a minta osztálydiagramját, valamint adja meg a mintában szereplő osztályok szerepét! (10p)

7) Csővezetékek és szűrők architektúra
   a) Ismertesse az adatforrás által vezérelt csővezeték architektúrát szekvenciadiagram segítségével (magyarázatot is adjon meg a diagramhoz)! (6p)
   b) Adjon meg pszeudo kódot szűrő (filter) komponens megvalósításához adatforrás által vezérelt csővezeték architektúrában! (6p)

8) Webalkalmazások (14p)
   a) Ismertesse 3-4 mondatban az ASP.NET kiszolgáló oldali vezérlők szerepét. (5p)
   b) Mutasson példát ASP.NET inline script alkalmazására, rövid magyarázattal! A HTML részeket is adja meg! (6p)

a) Mutasson egysoros példakódot arra, hogy lehet egyszerűen a kiszolgáló oldalon megoldani, hogy az egyes felhasználókhoz eltároljuk (a memóriában) az utolsó hozzáférésük idejét. Az aktuális idő a DateTime.Now tulajdonsággal érhető el. Azt feltételezheti, hogy minden hozzáférés alkalmával meghívódik a kiszolgálón egy Application_BeginRequest() { } függvény (a feladat tulajdonképpen a függvény törzsének a megírása). (3p)