**Web programozás és haladó fejlesztési technikák**

**2. zh minta**

(by: Kovács András)

A 2000-es években a Budapesti Műszaki Főiskolát megalapítják három főiskola összevonásával. Ezek a Kandó Kálmán Műszaki Főiskola, a Bánki Donát Műszaki Főiskola és a Könnyűipari Műszaki Főiskola. Az akkori informatikai integráció igen nehézkes volt. A zárthelyi dolgozatban demonstrálja, hogy jelen kor technikai lehetőségeivel hogyan lehetne **90 perc munkával** a három főiskola címtárát integrálni egy közös címtárba! Valósítson meg egy tárhely és egy email küldő szolgáltatást is!

Természetesen egy demonstráció akkor tökéletes, ha megfelelő teszteléssel is alá van támasztva! 😊

1. **Párhuzamos adatletöltés (14p)**

Készítsen egy statikus string tömböt a következő URL címekkel (kézzel írja bele):

* <http://users.nik.uni-obuda.hu/kovacs.andras/haladoprog/data/bmf.xml>
* <http://users.nik.uni-obuda.hu/kovacs.andras/haladoprog/data/kmf.xml>
* <http://users.nik.uni-obuda.hu/kovacs.andras/haladoprog/data/kimf.xml>

Készítsen ezután egy XDocument 3 elemű tömböt! Gondoskodjon arról, hogy a három URL címről letöltött adat, három XDocument reprezentációja, három független, párhuzamos szálon kerüljön be az XDocument tömbbe! A párhuzamosítást Task-ok segítségével valósítsa meg!

A párhuzamosított rész után készítsen egy újabb Task-ot, amely a három párhuzamosan futó Task elkészültekor indul! Használjon Continuationt! Ebben az újabb Task-ban hívja meg a következő feladatok függvényeit!

1. **Konvertálás (4p)**

Készítsen egy Dolgozo osztályt, amely név és email cím publikus, automatikus tulajdonságokkal rendelkezik!

Készítsen egy statikus függvényt, amely egy három elemű tömböt ad vissza. Minden tömbelemben Dolgozók listája található. A függvény bemenete a három elemű XDocument tömb!

1. **Egyesítés (4p)**

Készítsen egy statikus függvényt, amely az előző feladat visszatérési értékével dolgozik! Az egyesítés egy Dolgozo listával tér vissza, amelyben már @bmf.hu végű email címekkel szerepelnek a dolgozók! Hogyha egy ugyanolyan nevű dolgozó két főiskolán is dolgozott, akkor feltételezzük, hogy ő ugyanaz a személy. Az egyesített listába csak egyszer kerüljön bele!

1. **Samba szolgáltatás (3p)**

Egy statikus függvénnyel járja be a dolgozók listáját, és az email címük alapján hozzon nekik létre egy mappát! Directory.CreateDirectory() beépített függvénnyel dolgozzon (System.IO névtér!)

1. **Menü (2p)**

Jelenítsen meg egy menüt, amelyben fel vannak sorolva a dolgozók nevei. Ha kiválasztunk egyet, akkor emailt küldhetünk neki!

1. **Email szolgáltatás (3p)**

Készítsen egy új konzolos alkalmazást, amelynek parancssori paraméterként átadhatunk egy mappa elérési utat. Az alkalmazás megkérdezi az email címünk és az üzenetet, majd az adott mappába beszúr egy email\_cimunk.txt nevű fájlt, beleírja a pontos időt és az üzenetet! Ha van már ilyen fájl, akkor hozzáfűzést alkalmazunk!

1. **Email szolgáltatás meghívása (10p)**

A főprogramban a menüválasztás után Process segítségével meghívjuk a 6. feladatban készített alkalmazást! Várjuk meg, amíg lefut.

1. **Tesztelés (10p)**

A főprogram Egyesítő algoritmustát teszteljük NUnit frameworkkel! Szerepeljen egy előkészítő függvény, amely létrehozza a dolgozók listájának tömbjét. Ezenkívül 2 tesztet csináljunk. Az egyikkel ellenőrizzük, hogy a duplikált dolgozók valóban egyszer kerülnek csak be, a másikkal ellenőrizzük, hogy az email címek helyesen lettek-e létrehozva!

**Összesen elérhető: 50 pont**