**Web programozás-és haladó fejlesztési technikák**

Első zh-ra felkészülő feladatsor

(by: Kovács András)

Önnek egy olyan szoftvert kell készítenie, amely két adatforrásból dolgozik. A megkapott adatokat egyesítve kell kimutatásokat végeznie, valamint a kimutatások megjelenítésére egy adattípus független függvényt kell készítenie!

1. **Kurzusok listája (0p)**

Hozzon létre egy Service-Based Database elemet a Solution-ben, amelybe hozzon létre egy táblát a 2. sz. melléklet SQL scriptjével!

1. **Hallgatók adatai (0p)**

Az 1. sz melléklet XML adatait illessze be egy megfelelő .xml fájlba!

1. **Student osztály (0p)**

Hozzon létre egy Student osztályt az alábbi automatikus tulajdonságokkal!

* Name (string)
* Course (string)
* Mark (int)
* Type (string)
1. **XML ből gyűjtemény (5p)**

Készítsen a Student osztályba egy statikus függvényt, amely képes List<Student> formában visszaadni az XML fájl tartalmát!

1. **Entity Framework (2p)**

Üzemelje be az Entity Frameworköt, az adatbázis táblát érje el egy közvetítő osztályon keresztül a tanult módon!

1. **LinqResultConsole(15p)**

Írjon egy statikus osztályt LinqResultConsole néven, ami egy statikus DisplayResult<T>( IEnumerable<T> result) metódussal rendelezik!

Ennek a függvénynek bármilyen linq lekérdezés eredményét közvetlenül át tudja adni. A gyűjtemény típusát a T generikus paraméterből kapja vissza. (segítség: Type t = typeof(T)). A gyűjteményt magát a result paraméterből éri el.

Ez a függvény megvizsgálja, hogy az adott típus, amelyből gyűjteményt kap, milyen tulajdonságokkal rendelkezik. A tulajdonságok neveit írja ki, mint egy táblázat fejlécsorát. Ezután a tulajdonságok ilyen sorrendjében írja ki külön sorokba a gyűjtemény egyes elemeinek értékeit!

Fel kell készülni arra is, hogy adott Linq lekérdezés eredményeként IEnumerable<int> vagy IEnumerable<string> gyűjteményt kap. Ezeknek nincsenek további tulajdonságai. Kezelje le, hogy int és string gyűjtemény esetén csak a gyűjteményt írja ki a metódus, fejléc nélkül!

1. **Neptun osztály (0p)**

Készítsen egy Neptun osztályt, amely konstruktorában vegye át a hallgatók listáját, valamint az adatbázis kapcsolati osztályt! Az osztály rendelkezzen az alábbi metódusokkal! Minden metódus void, a linq lekérdezés eredményét a LinqResultConsole-nak adja át, ami megjeleníti!

1. **Opre hallgatók (2p)**

Az operációs rendszerek tárgyat tanuló hallgatók minden adatát szeretnénk megkapni!

1. **Opre hallgatók neve (2p)**

Az operációs rendszerek tárgyat tanuló hallgatók névsorát szeretnénk megkapni!

1. **Tantárgy átlagok rövid nevekkel (5p)**

Minden tantárgy rövid nevét (ami a Student osztályban van) írjuk ki, továbbá azt, hogy az adott tárgyon milyen átlageredmény született!

1. **Tantárgy átlagok hosszú nevekkel (7p)**

Minden tantárgy hosszú nevét (ami az adatbázisban van) írjuk ki, továbbá azt, hogy az adott tárgyon milyen átlageredmény született!

1. **Tantárgyak és tanulók (6p)**

Írjuk ki a tanulók neveit, és mellé, hogy milyen tantárgyat hallgattak! A tantárgy hosszú neve szerepeljen itt!

1. **Konzol színezés (6p)**

Készítsen egy ColoredPropertyAttribute osztályt, amely egy ConsoleColor-t tartalmaz tulajdonságként. Hogyha a Student osztályban ezt az attribútumot rátesszük néhány tulajdonságra, akkor a LinqResultConsole vegye figyelembe az itt megadott színértékeket, és ha olyan gyűjteményt kap, amelynek típusán vannak ilyen attribútumok, akkor az attribútumban található színnel írja a konzolra!

**összesen: 50 pont**

**melléklet 01: students.xml tartalma**

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<students>

 <student>

 <name>Erdei Gréta</name>

 <course>halal</course>

 <mark>3</mark>

 <type>esti</type>

 </student>

 <student>

 <name>Huszár Róbert</name>

 <course>fnyg</course>

 <mark>3</mark>

 <type>nappali</type>

 </student>

 <student>

 <name>Kiss Géza</name>

 <course>sztf1</course>

 <mark>5</mark>

 <type>nappali</type>

 </student>

 <student>

 <name>Kiss Géza</name>

 <course>archi1</course>

 <mark>4</mark>

 <type>nappali</type>

 </student>

 <student>

 <name>Kiss Géza</name>

 <course>whp</course>

 <mark>4</mark>

 <type>nappali</type>

 </student>

 <student>

 <name>Kiss Miklós</name>

 <course>archi1</course>

 <mark>2</mark>

 <type>esti</type>

 </student>

 <student>

 <name>Kiss Miklós</name>

 <course>opre</course>

 <mark>2</mark>

 <type>esti</type>

 </student>

 <student>

 <name>Nagy Attila</name>

 <course>archi1</course>

 <mark>5</mark>

 <type>nappali</type>

 </student>

 <student>

 <name>Nagy Attila</name>

 <course>opre</course>

 <mark>5</mark>

 <type>nappali</type>

 </student>

 <student>

 <name>Nagy Attila</name>

 <course>halal</course>

 <mark>5</mark>

 <type>nappali</type>

 </student>

 <student>

 <name>Nagy Ferenc</name>

 <course>whp</course>

 <mark>2</mark>

 <type>esti</type>

 </student>

 <student>

 <name>Nagy Ferenc</name>

 <course>sztf1</course>

 <mark>3</mark>

 <type>esti</type>

 </student>

 <student>

 <name>Péter Mónika</name>

 <course>archi1</course>

 <mark>2</mark>

 <type>nappali</type>

 </student>

 <student>

 <name>Péter Mónika</name>

 <course>opre</course>

 <mark>1</mark>

 <type>nappali</type>

 </student>

 <student>

 <name>Pollák Csilla</name>

 <course>opre</course>

 <mark>4</mark>

 <type>esti</type>

 </student>

 <student>

 <name>Szôllôssy Janka</name>

 <course>whp</course>

 <mark>5</mark>

 <type>esti</type>

 </student>

 <student>

 <name>Szôllôssy Janka</name>

 <course>fnyg</course>

 <mark>1</mark>

 <type>esti</type>

 </student>

 <student>

 <name>Tóth László</name>

 <course>whp</course>

 <mark>4</mark>

 <type>esti</type>

 </student>

</students>

**melléklet 02: SQL script**

**IF** OBJECT\_ID('course', 'U') **IS** **NOT** **NULL** **DROP** **TABLE** course;

**CREATE** **TABLE** course (

    courseId **NUMERIC**(3) **NOT** **NULL**,

    courseShortName **VARCHAR**(200),

    courseLongName **VARCHAR**(200),

    credit **INT** **NOT** **NULL**

    **CONSTRAINT** COURSE\_PRIMARY\_KEY **PRIMARY** **KEY** (courseId));

**INSERT** **INTO** course **VALUES** (1,'sztf1','Szoftver tervezés-és fejlesztés I.',6);

**INSERT** **INTO** course **VALUES** (2,'opre','Operációs rendszerek',5);

**INSERT** **INTO** course **VALUES** (3,'whp','Web programozás-és haladó fejlesztési technikák',4);

**INSERT** **INTO** course **VALUES** (4,'archi1','Számítógép architektúrák alapjai I.',2);

**INSERT** **INTO** course **VALUES** (5,'halal','Haladó algoritmusok',3);

**INSERT** **INTO** course **VALUES** (6,'fnyg','Formális nyelvek és gépek',2);