**Feladat: "Employee" model és repository készítése**

**Cél:**

Készítsd el az Employee modellt és a hozzá tartozó repository osztályt! A modellt és az adatokat egy SQLite adatbázisban fogjuk tárolni, amelyet a dolgozok.csv fájl alapján scaffoldolunk.

**1. Adatok előkészítése**

A tesztadatokat egy CSV fájl tartalmazza (dolgozok.csv):

email,name,salary

kovacs.peter@ceg.hu,Kovács Péter,450000

nagy.anna@munkahely.com,Nagy Anna,520000

szabo.mate@ceg.hu,Szabó Máté,480000

toth.eva@vallalat.net,Tóth Éva,510000

**2. Employee modell létrehozása**

Hozz létre egy Employee osztályt, amely tartalmazza az alábbi elemeket:

* **Privát változók**: Két privát változó, amely a fizetést és az email címet tárolja.
* **Konstruktor**: Az osztály konstruktorában az email cím és a név inicializálása történik, kivételkezeléssel ha azok üresek.
* **Publikus tulajdonságok**: A név lekérdezhető és módosítható, az email cím és a fizetés csak a privát változókon keresztül érhető el.
* **Adókulcs konstans**: Egy konstans érték, amely az adókulcsot tárolja.
* **Adó összege**: Egy csak lekérdezhető tulajdonság, amely kiszámolja a fizetésre vonatkozó adó összegét.
* **Fizetés számláló**: Egy változó, amely számolja, hogy a dolgozó hányszor kapott fizetést.
* **Metódus a fizetés növelésére**: Csak pozitív értékkel lehessen növelni a fizetést, ellenkező esetben kivételt dobjon, valamint növelje a fizetés számlálót.
* **ToString metódus**: Formázott szöveget ad vissza a névvel, email címmel, jelenlegi fizetéssel és a fizetések számával.

**Elvárt konzol kimenet Nagy Anna példáján:**

Név: Nagy Anna

Email: nagy.anna@munkahely.com

Fizetés: 520000 Ft

Adó: 104000 Ft

Fizetési ciklusok száma: 0

**3. Model réteg tesztelése**

Az elkészült Employee osztály ellenőrzésére készíts egy tesztet a Program.cs fájlban az alábbi lépésekkel:

* Hozz létre egy Employee példányt üres email címmel, és ellenőrizd, hogy kivételt dob-e.
* Hozz létre egy Employee példányt "nagy.anna@munkahely.com" email címmel és "Nagy Anna" névvel, majd írasd ki a konzolra az adatait.
* Próbálj meg -50000 összeget hozzáadni a fizetéshez, és ellenőrizd, hogy kivételt dob-e.
* Adj hozzá 25000 és 40000 összeget a fizetéshez, majd írasd ki az objektumot a konzolra.
* Ellenőrizd az adókulcsot és az adó összegét.
* Ellenőrizd, hogy a fizetési ciklusok száma helyesen növekszik-e.

**4. EmployeeContext**

* Scaffoldold az adatbázist a dolgozok.csv fájl alapján.
* A generált DbContext osztályban használd az Employee osztályt.
* Az email mező legyen kulcs!
* Biztosítsd, hogy a salary mező egész számként (INT) legyen tárolva az adatbázisban.

**5. Repository osztály létrehozása**

Hozz létre egy EmployeeRepo osztályt, amely felelős az adatbázis műveletekért!

Feladatok és elvárt konzol kimenetek:

* **Az adatbázisban lévő dolgozók számának lekérése**
  + SQL: SELECT COUNT(\*) FROM Employees;
  + **Elvárt kimenet:** A dolgozók száma: 4
* **Minden dolgozó listázása (Név - Email - Fizetés)**
  + SQL: SELECT name, email, salary FROM Employees;
  + **Elvárt kimenet:**

Kovács Péter (kovacs.peter@ceg.hu) - 450000 Ft

Nagy Anna (nagy.anna@munkahely.com) - 520000 Ft

Szabó Máté (szabo.mate@ceg.hu) - 480000 Ft

Tóth Éva (toth.eva@vallalat.net) - 510000 Ft

* **Adott összeg felett keresés (pl. 480000 Ft)**
  + SQL: SELECT name, email, salary FROM Employees WHERE salary > ?;
  + **Elvárt kimenet:**

Nagy Anna (nagy.anna@munkahely.com) - 520000 Ft

Tóth Éva (toth.eva@vallalat.net) - 510000 Ft

* **Dolgozók rendezése fizetés szerint csökkenő sorrendben**
  + SQL: SELECT name, email, salary FROM Employees ORDER BY salary DESC;
  + **Elvárt kimenet:**

Nagy Anna (nagy.anna@munkahely.com) - 520000 Ft

Tóth Éva (toth.eva@vallalat.net) - 510000 Ft

Szabó Máté (szabo.mate@ceg.hu) - 480000 Ft

Kovács Péter (kovacs.peter@ceg.hu) - 450000 Ft

* Teljes bérkeret és átlagbér kiszámítása
  + SQL: SELECT SUM(salary), AVG(salary) FROM Employees;
* Dolgozók csoportosítása fizetés szerint és listázásuk
  + SQL: SELECT salary, COUNT(\*) FROM Employees GROUP BY salary;
  + Csoportok:
    - 400000 Ft alatt
    - 400000 - 500000 Ft között
    - 500000 Ft felett
* Dolgozók adójának kiszámítása és listázása
  + SQL: SELECT name, salary \* 0.2 AS tax FROM Employees;

Kovács Péter - Adó: 90000 Ft

Nagy Anna - Adó: 104000 Ft

Szabó Máté - Adó: 96000 Ft

Tóth Éva - Adó: 102000 Ft

* Hányszor kapott fizetést a dolgozó adat tárolása és lekérdezése
  + SQL: SELECT name, pay\_cycles FROM Employees;
* Új dolgozó hozzáadása (csak ha az email cím még nem létezik)
  + SQL: INSERT INTO Employees (email, name, salary) SELECT ?, ?, ? WHERE NOT EXISTS (SELECT 1 FROM Employees WHERE email = ?);
* Adott dolgozó fizetésének módosítása
  + SQL: UPDATE Employees SET salary = ? WHERE email = ?;
* Dolgozó törlése (csak ha az email cím létezik)
  + SQL: DELETE FROM Employees WHERE email = ?;

Az eredményeket jelenítsd meg a konzolon az egyes metódusok kipróbálásával!