**Dolgozókat kezelő alkalmazás fejlesztése**

**Feladat**

Készítsen egy dolgozókat kezelő rendszert, amely egy konzolos és egy asztali alkalmazásból áll. A rendszer célja a dolgozók adatainak kezelése, fizetések kiosztása és az adatok megjelenítése.

**Konzolos alkalmazás**

A konzolos alkalmazás felelős a dolgozók adatainak tárolásáért és kezeléséért. Az adatokat egy CSV fájlból olvassa be, és egy adatbázisban tárolja.

**Fejlesztendő funkciók:**

*Dolgozó létrehozása és adatok validálása kifejlesztendő osztály:*

* A dolgozónak van neve, email címe és pillanatnyi fizetése.
* A dolgozó email címe és fizetése az osztályban védett adatként is kezelhető.
* Dolgozó létrehozásakor a dolgozó email címét és nevét kell megadni, a fizetése kezdetben nulla.
* A dolgozó létrehozásakor ellenőrizni kell, hogy az email és a név mezők nem lehetnek üresek, amennyiben üres adat kerül megadásra, kivételt kell dobni.
* A dolgozó email címének megadásakor ellenőrizni kell, hogy "@" jelet tartalmaz-e!

*Dolgozókkal kapcsolatos üzleti logika:*

* A dolgozó számára fizetést lehet kiosztani.
* Negatív vagy nulla összeg esetén kivételt kell dobni.
* Több fizetést is kaphat, az összeg összeadódik.

*Dolgozók kezelésére kifejlesztendő osztály:*

* Az adatok kezelése egy adatbázisban, amely a salary.csv fájlból importálja az adatokat.
* Dolgozók számának meghatározása.
* Fizetést kapó és fizetés nélküli dolgozók számának meghatározása.
* A dolgozók által kapott teljes fizetések átlaga.
* A legnagyobb és legkisebb fizetést kapó dolgozó nevének meghatározása.
* Email domainnevenként a dolgozók számának meghatározása.
* Adott email címmel rendelkező dolgozó fizetést kap.
* Adott email címmel rendelkező dolgozó, aki fizetés nélkül van, törlése.
* Új, nulla fizetéssel és az adatbázisban nem létező email címmel rendelkező dolgozó adatának felvétele (a dolgozó nevét is meg kell adni).
* Dolgozók szűrése névrészlet alapján.
* Dolgozók szűrése domain nevek alapján.
* Dolgozók szűrése adott összeg közötti fizetéssel.

**Teszteléshez használandó kód:**

Console.WriteLine("1. feladat");

try

{

Employee empty = new Employee("", "Null");

}

catch(Exception e)

{

Console.WriteLine(e.Message);

}

Console.WriteLine("2. feladat");

Employee janos = new Employee("janos@ceg.hu", "Dolgozó János");

Console.WriteLine(janos);

Console.WriteLine("3. feladat");

try

{

janos.Pay(-300000);

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

Console.WriteLine("4. feladat");

janos.Pay(200000);

janos.Pay(300000);

Console.WriteLine(janos);

EmployeeRepo repo = new EmployeeRepo();

Console.WriteLine($"Dolgozók száma: {repo.GetNumberOfEmployees()}");

**Asztali alkalmazás**

Az asztali alkalmazás a konzolos alkalmazás adatbázisát használja, és egy felhasználóbarát felületen keresztül biztosítja az adatok megjelenítését és kezelését.

**Fejlesztendő funkciók:**

* A vezérlőpulton jelenjen meg:
  + Az összes dolgozó száma.
  + Fizetést kapó és fizetés nélküli dolgozók száma.
  + A legnagyobb és legkisebb fizetést kapó dolgozó neve.
  + A dolgozók átlagkeresete.
  + Email domainnevenként a dolgozók száma.
* A dolgozók kezelésének lehetőségei:
  + Összes dolgozó adatának megjelenítése.
  + Adott email címmel rendelkező dolgozó fizetést kap.
  + Adott email címmel és fizetéssel nem rendelkező dolgozó törlése.
  + Új, nulla fizetéssel és az adatbázisban nem létező email címmel rendelkező dolgozó adatának felvétele (a dolgozó nevét is meg kell adni).
  + A dolgozók adatának szűrése névrészlet alapján.
  + A dolgozók szűrése domain nevek alapján.
  + Dolgozók szűrése adott összeg közötti fizetéssel

A fejlesztés során ügyeljenek a tiszta kód alapelveire, és megfelelően szervezzék az alkalmazás rétegeit a konzol és az asztali alkalmazás között.