## A feladat összegzése

* olyan blog alkalmazás készítése, amelyben a regisztrált felhasználók blogokat hozhatnak létre maguknak sablonok alapján, a blogokhoz bejegyzéseket írhatnak, és kommentelhetnek bejegyzésekre vagy már meglévő kommentekre (akár a sajátjukra is)
* az alkalmazásban regisztráció nélkül is lehet blogbejegyzéseket olvasni, viszont azokat írni vagy kommentelni nem
* az alkalmazásban a regisztrált felhasználók szerepkörrel rendelkeznek
* a szerepkör lehet: admin / moderator / user
* a szerepkörökhöz tartozó jogosultságok megírása a feladat része
  + *példák lehetséges jogosultságokra (nem kötelezendően megvalósítandó):*
    - *felhasználók adatainak olvasása*
      * *csak admin, illetve a saját felhasználói adataihoz minden user hozzáfér*
    - *felhasználók adatainak módosítása*
      * *csak admin, illetve a saját felhasználói adatait minden user módosíthatja*
    - *blogbejegyzések olvasása*
      * *nem kell hozzá jogosultság, sőt regisztráció sem*
    - *blogbejegyzések módosítása*
      * *admin, moderator; illetve a saját blogbejegyzéseit minden user módosíthatja*
    - *blogbejegyzések törlése*
      * *admin, moderator; illetve a saját blogbejegyzéseit minden user törölheti*
    - *kommentek írása*
      * *admin, moderator, user*

## Általános elvárások

* **a feladat beadható a szükséges fájlokat tartalmazó tömörített mappában, vagy GitHub-ra feltöltött (publikus) repository-ként, a repository linkjének elküldésével**
* az alkalmazás megjelenítése, azaz (grafikus) felhasználói felület létrehozása **NEM** része a feladatnak
* a program forráskódjának **nem kell main() metódust tartalmaznia**, de igény szerint futtathatónak kell lennie - azaz **szintaktikailag helyesnek** kell lennie
* az adatbázis létrehozásához szükséges script-nek szintén futtathatónak kell lennie
* a blog-sablonok ne legyenek módosíthatóak (azaz nem kell gondoskodni arról, hogy a sablonból létrehozott blog eltérő színű, betűtípusú stb., mint az eredeti sablon)
* az egyszerűsítés kedvéért: **a jelszavak titkosítása nem a feladat része**, azonban tudd, hogy a későbbiekben szükséges (tehát jelen feladatban - **csak és kizárólag jelen feladatban** - megengedett a jelszavak titkosítás nélküli tárolása az adatbázisban)

## Feladatok

### Adatbázis tervezése és létrehozása

A feladat teljesítéséhez leadandó:

* adatbázis tervének diagramja, amely tartalmazza a táblákat, a táblák mezőit és a táblák közötti kapcsolatokat
* adatbázis létrehozásához szükséges SQL script

Az adatbázis a következő paramétereknek feleljen meg (ezeken túl igény szerint bővíthető):

* + - tárolja a regisztrált felhasználók adatait
    - tárolja a választható blog-sablonokat (amik alapján a felhasználók elkészíthetik a saját blogjukat)
      * *megjegyzés:* ***nem kell teljes blogsablonokat létrehoznod****… elegendő pl. a blogsablon nevét, kategóriáját, fő színét, háttérképét tárolnia az adatbázisnak*
  + tárolja a felhasználók blogjait
    - lehetőséget biztosít arra, hogy egy felhasználó több blogot is létrehozzon
  + tárolja a blogokhoz tartozó bejegyzéseket
    - lehetőséget biztosít a blogbejegyzések állapotának megadására (ami lehet pl. közzétett, piszkozat stb.)
    - lehetőséget biztosít arra, hogy egy bloghoz több blogbejegyzés is tartozhasson
  + tárolja a blogokhoz tartozó kommenteket
    - lehetőséget biztosít arra, hogy egy blogbejegyzéshez több komment is tartozhasson
    - a komment lehet a bejegyzésre közvetlenül adott reakció, de
    - lehet egy másik kommentre adott válasz

### Java program forráskódjának megírása

A feladat teljesítéséhez leadandó:

* a főbb osztályok vázlatos diagramja, amelyben szerepel az osztályok neve, néhány szavas leírása, és az esetleges kapcsolatok az osztályok között (egyértelműen jelölve)
  + főbb osztályok pl. a modellek, kontrollerek, szervizek stb. - a kisebb segédosztályok, mint amilyenek pl. a helperek, nem sorolandók a főbb osztályok közé
  + kapcsolatok alatt értendő pl. öröklődés, kompozíció - a jelölésből derüljön ki, hogy miről van szó (pl. jelmagyarázattal)
* a program forráskódja maga
  + minden, ami szükséges a program futtatásához (osztályok, package-ek, resources stb.)

A program forráskódja az alábbi paramétereknek feleljen meg:

* strukturált, logikusan rendezett és szerkesztett elrendezésű (követi a programfelépítés alapelveit, az OOP koncepcióit)
* átlátható, olvasható és értelmezhető (követi a forráskód írásának alapelveit)
* szintaktikailag helyes
* következetes
* tartalmazza az adatbázisban tárolt adatok feldolgozásához szükséges osztályokat és metódusokat
* tartalmazza az adatbázis-kapcsolathoz szükséges eszközöket (osztályokat, metódusokat)
  + JDBC-n keresztül
  + ***a saját adatbázis-szervered kapcsolódásához szükséges felhasználónevet és jelszót szigorúan tilos beleírnod a forráskódba!***
* lehetőséget biztosít a felhasználói szerepkörök és jogosultságaik kezelésére
* tartalmaz egy-egy metódust, amely teljesíti az alábbi feladatokat (egy feladat - egy metódus):
  + visszaadja adott szerepkörű felhasználók összes adatát
  + visszaadja egy adott felhasználó összes adatát
  + visszaadja egy adott felhasználó összes blogját
  + visszaadja egy adott blog összes blogbejegyzését
  + visszaadja egy adott blogbejegyzéshez tartozó összes kommentet

### Unit tesztek írása

A program forráskódja mellett, azonban a forráskódtól elkülönítve szerepeljenek Unit tesztek, amelyek tesztelik a logikát végző metódusok helyes működését, méghozzá több esetre is.

**Általában** nem logikát végző metódusok, ezekhez nem szükséges tesztek írása: konstruktorok, getterek, setterek. (De nem hiba a tesztelésük!)