

Jussunk el az előző gyakorlat végére. Ez a leglényegesebb:

Tartalomjegyzék

- 1 5. Felhasználó rendszer
- 2 Feladatok
 - ◆ 2.1 User kiegészítése
 - ◆ 2.2 Felhasználó rendszer kiegészítése

5. Felhasználó rendszer

Írjatok egy osztályt (ti találjátok ki a nevét, de legyen beszédes és célratörő, általában a programozásban ez a legnehezebb feladat) ami a korábban megírt **User** objektumokból tud sokat tárolni (tömb). Továbbá valamilyen módon tárol még kommenteket is. Egy komment az üzenetből és a készítőjéből áll, aki egy **User**. Találjátok ki, hogyan tárolnátok a kommenteket, érdemes új osztályt bevezetni rá. A rendszernek magának legyen 2 függvénye, egyik amivel új **User**t lehet felvenni és egy amivel új kommentet lehet írni.

Pár tanács:

- Hogy a tömb lefoglalással ne legyenek gondok, amikor létrejön ez a rendszer foglaljatok le mondjuk 500 **User**nek helyet és egy számlálóval mentsetek, hogy pontosan mennyi is van valóban a rendszerben.
- Az előbbihez az kell, hogy a **User** osztálynak legyen **default konstruktora**, csináljatok hát neki.
- A **User** felvétele függvény talán úgy a legegyszerűbb, ha egy bemenete van és az a **User** objektum, amit be akarunk rakni a tömbbe.
- A komment tárolásánál elég ha beküldő nevét tároljátok mondjuk, itt nem kell teljes **User**t tárolni.
- Mivel kommentből is sokat kell tárolni, ezért nagyon hasonló módon meg lehet oldani a tárolásukat, mint a **Userekét**.

Feladatok

A következő két feladatot érdemes lehet párhuzamosan oldani.

User kiegészítése

- Legyen jelszóváltoztatás, amihez 2 bemenet, régi jelszó, új jelszó. Használjuk a **passwordCheck** metódust.
- Legyen a usereknek jogosultságuk, ez lehet mondjuk egy String. Hogy milyen jogosultságok vannak, majd később mondjuk meg.
- Hozzatok létre egy új osztályt **PersonalInfo** néven, legyen inkább a teljes név ebben tárolva, és a **User**nek legyen egy ilyen adattagja. Így sokkal könnyebben tudunk majd új adatokkal kiegészíteni a **User**ünket. Azért rakjunk is bele még pár dolgot, pl születési évet. Vegyük észre, hogy emiatt módosítani kell a **User** konstruktorát is.
- A **PersonalInfo**nak legyen minden adattagjához egy set függvénye. Pl: **setBirthYear**.
- Ezeket delegáljuk a **User**nek, eclipse-ben, jelöljük ki a **PersonalInfo** adattagját a **User**nek felül Source -> Generate Delegate methods.
- Legyen egy statikus adattagja a **User**nek, ami alaphól 0 és minden konstruktor hívásnál növeljük eggyel.

Felhasználó rendszer kiegészítése

- A **User** kiegészítése miatt elképzelhet?, hogy elromlott ez az osztály. Javítsátok meg.
- Írjatok egy metódust amivel szépen ki tudjátok iratni a kommenteket, legyen ott az is, hogy ki írta.
- Legyen egy olyan metódus, ami csak egy adott user kommentjeit írja ki.
- Tanács: a kommentek osztálynak írjatok egy kiíró metódust, hogy majd csak ezt kelljen meghívni.