

1.

Készítsünk el egy kliens-szerver alkalmazást az alábbi problémára:

A szervernek kezelnie kell felhasználókat <User>, motorok egy listáját, a felhasználók motorokat kölcsönözhetnek.

A felhasználók adatai:

név <string>

jelszó <string>

A felhasználók adatait egy `userek.txt` file tartalmazza.

A fájlban egy felhasználó adatai egy sorban vannak leírva, <nev>|<jelszo>alakban.

A szerver induláskor olvassa fel az adatokat! A motorok adatait egy `motorok.txt` fájl tartalmazza, egy motor adatai egy sorban, | jellel elválasztott formában. (rendszám, típus, km, kölcsönzési díj km-enként).

A szerver induláskor olvassa fel a fájl tartalmát!

A szervernek támogatni kell az alábbi funkciókat:

HILFE: kilistázza az elérhető parancsok neveit és funkcióit, login nélkül is elérhető.

LISTA : motorok listázása, login nélkül is elérhető.

LOGIN|<nev>|<pass>: Bejelentkezteti az adott felhasználót.

LOGOUT: kijelentkezteti az adott klienst a szerverről, ha az bejelentkezett!

KOLCSONZES|<rendszam>: A választott rendszámú motor kikölcsönzése. Értelmszerűen ellenőrizni kell, hogy a motor nincs jelenleg kikölcsönözve, és csak login után adhatjuk ki.

VISSZAVISZ|<rendszam>|<km>: Jelezzük, hogy a választott rendszámú motort visszavisszük. Csak login után adható ki. Az új km-t rögzíteni kell, nem lehet kisebb, mint a korábbi. Válaszként vissza kell adni a kölcsönzés teljes díját.

EXIT: kommunikáció befejezése, kliens kilép.

A kliens és a szerver kódját is el kell készíteni! Az üzenetek stream-ekben "közlekedjenek". Egyszerre több klienssel tudjon a szerver kommunikálni! A szerver ne álljon le, ha a kliens bontja a kapcsolatot!

Egy ZIP fájlt kell feltölteni!

<https://oc.uni-eszterhazy.hu/s/tJVd9q6nyJOKroO>