

FONTOS: a feladatok leírása egy alapnak tekinthető. Egyes részproblémák megoldásához nyugodtan változtathatsz minimális szinten ott, ahol szükségesnek érzed.

1. FELADAT – Interface használata mint „felület”

IHallgatóFelület

- Olvasható név
- Olvasható neptunkód
- Írható és olvasható tetszám
- Tárgylistázás metódus (void, nincs paraméter)

IOktatóFelület

- Olvasható neptunkód
- Olvasható név
- Írható és olvasható Programozás jegy

ITOFelület

- Írható és olvasható neptunkód, telefonszám, név
- Írható és olvasható tárgyak tömbje
- Olvasható tárgyak darabszáma

Hallgató

- Valósítsa meg a három interface-t
- A szükséges adatmezőkkel egészítsd ki az osztályt
- Konstruktor, amely a tárgyak darabszámát fogadja (tárgyak tömb mérete)
- Tárgyfelvesz metódus (Targy targy)

Tárgy

- Név
- Kreditérték
- Érdemjegy
- toString override
- Legyen összehasonlítható (IComparable)

Csinálj egy **Neptun** osztályt, amiben hallgató típusú elemek vannak egy tömbben. A konstruktorban lehessen megadni, hogy hány elemű legyen ez a tömb.

HallgatóFelvitele metódus: amely a TO-n van, és egy hallgatót tudnak felvinni a rendszerbe. Minden adatot meg tudnak adni a hallgatónak, és ez bekerül a neptun-on belüli gyűjteménybe.

Jegybeírás metódus: Oktatóknak szóló funkció, egy hallgatónak jegyet tudnak beírni.

TelefonszámMódosítás metódus: egy hallgató tudja módosítani magának a telefonszámát.

Main-be csinálj egy neptun példányt és rajta keresztül hívd/teszteld le a metódusokat.

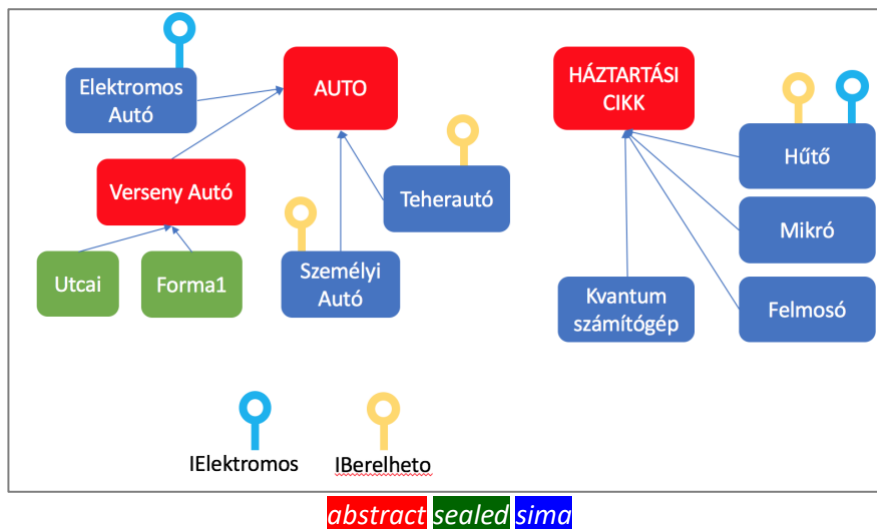
2. FELADAT – Interface mint kötelezően megvalósítandó metódusok/tulajdonságok előírása

IElektromos

- Int fogyasztás();
- Int töltési idő();

IBerelhető

- Int olvasható és írható **bérleti díj**
- int olvasható és írható **bérlési napok száma** (hány napra van bérbe adva)
- bool olvasható és írható **bérlés aktív** (jelenleg bérelve van/nincs)



Kölcsönző osztály

- Felvesz metódus, amely egy bérelhető típust vesz fel egy gyűjteménybe. Ennek hatására ez az eszköz bérelhető lesz a kölcsönzőből.
- BérbeAd metódus, amely az adott terméket beállítja, hogy annak bérlése jelenleg aktív. Tehát más nem tudja kibérelni.
- VisszaHoz metódus, amely a bérbeadásból visszahozott terméket újra bérbeadhatónak jelöli.
- BérbeadásSzámológ metódus, amely az egyes termékeket számolja, hogy hányszor voltak bérbe adva (az ezt nyilvántartó gyűjtemény ugyan annyi elemű legyen, amennyi a bérbeadható termékek gyűjtemény elemszáma)
- SorrendbeRak metódus, amely a bérelhető termékeket csökkenő sorrendbe rendezi. Ennek megvalósításához használd az IComparable interfészt a szükséges osztályokon. A sorrend alapját a bérleti díj képezze.

3. FELADAT – prog2 diából

http://users.nik.uni-obuda.hu/prog2/Prog2_labor_1.pdf

Saját interfészek

- **Feladat**
Valósítsuk meg az ábrán látható interfész és osztály hierarchiát
- Készítsünk főprogramot ami egy közös tömbben tárol **IFizetoEszkoz** interfészt megvalósító objektumokat
- Legyen egy fizetés metódusa, ami egy ilyen tömböt kap paraméterként, és akkor ad vissza igazat, ha bármelyik elem végre tudta hajtani a fizetést
- Legyen egy ellenőrzött fizetés metódus, ami hasonló, de a fizető nevét is ellenőrzi

