**OOP feladatok (python)**

1. Írj osztályt, ami egy ***Macska***objektumot valósít meg.  
   Minden metódus írja ki a konzolra amit csinált! Pl. ha evett a macska akkor írjuk ki: xy Macska evet/xy Macsak nem evett mert nem volt éhes.
   1. A macska adattagjai a következőek legyenek: név, súly, éhes -e.
   2. Készítsd el a konstruktorát amely paraméterlistájából kapja, az értékeket.
   3. Az osztálynak legyen egy *eszik* metódusa, ami egy float értéket vár (étel mennyisége). Boolean-el tér vissza (sikeres volt -e az etetés). Ha a macska éhes, az etetés sikeres, és a súlya nőjön az étel mennyiségével. A macska ezután ne legyen éhes. Ha a macska nem éhes, az etetés nem sikeres.
   4. Az osztálynak legyen egy *futkos* metódusa, ami nem vár paramétert. A macska súlya csökkenjen 0.1-el, és ha nem volt éhes, akkor éhezzen meg.
   5. Az osztálynak legyen egy *jelenlegiErtekek* metódusa ami kiírja a macska aktuális értékeit: név, súly, éhség állapot. Az éhség állapotra igen/nem-et írjon.
   6. Hozz létre két macskát a két különböző, és próbáld meg megetetni őket. Az etetés sikerességéről írj információt konzolra.
   7. Mindkét macskát etesd meg, majd futkosson.
   8. A legvégén írd ki az értékeit a *jelenlegiErtekek* metódussal.
2. Írj osztályt, ami egy ***Szamitogep***objektumot valósít meg.  
   Minden metódus írja ki a konzolra, amit éppen elvégzett, pl. Ha bekapcsolt akkor írja ki, hogy bekapcsolt.
3. A számítógép adattagjai a következőek legyenek: szabad tárhely MB-ban (float), be van -e kapcsolva (boolean), azonosító szám (string).
4. Készíts konstruktort is az osztályhoz, minden adattagot a paraméterlistából állít be.
5. Az osztálynak legyen, egy *kapcsol* metódusa, ami nem vár paramétert. Ha a gép ki van kapcsolva, akkor kapcsolja be, egyébként kapcsolja ki.
6. Az osztálynak legyen egy *programMasol* metódusa, ami egy program méretét várja paraméternek MB-ban (float). Ha a program ráfér még a gépre, és a gép be van kapcsolva, úgy csökkenjen a szabad memória a program méretével. A metódus térjen vissza boolean változóval, hogy sikeres volt-e a másolás (true)/nem volt (false)
7. Hozz létre két számítógépet a fenti osztállyal. Mindkét gép kikapcsolt állapotban kezdjen. Az első gépet kapcsold be, és másold rá először 800, aztán 400 MB programot. A másik gépre másolj 1 MB programot.
8. Írj osztályt, ami egy ***Hallgato*** objektumot valósít meg.  
   Minden metódus írja ki a konzolra, amit éppen elvégzett, pl. Ha bekapcsolt akkor írja ki, hogy bekapcsolt.
   1. A hallgató adattagjai a következőek legyenek: azonosító (String), évfolyam (int), kreditszám(int).
   2. Készíts konstruktort is az osztályhoz, minden adattagot a paraméterlistából állít be.
   3. Az osztálynak legyen egy *targyFelvesz* metódusa, amivel egy int paramétert (tárgykreditértéke) kér. A hallgató kreditszáma nőjön a kapott értékkel.
   4. Az osztálynak legyen egy boolean-al visszatérő *vizsgazik* metódusa, ami nem vár paramétert. Ha a hallgatónak 0-nál több kreditje van, akkor a sikeres a vizsga: a következő évfolyamba lép, és nullázódik a kreditszáma. Egyébként a vizsga sikertelen.
   5. Hozz létre két hallgatót. Az egyikük vegyen fel tárgyat, majd vizsgáztasd őket. A vizsga sikerességéről írj információt konzolra.
9. Írj osztályt, ami egy ***Torta*** objektumot valósít meg.  
   Minden metódus írja ki a konzolra, amit éppen elvégzett, pl. Ha bekapcsolt akkor írja ki, hogy bekapcsolt.
   1. A torta adattagjai a következőek legyenek: emeletek száma (int), meg van -e kenve krémmel (boolean).
   2. Készíts konstruktort is az osztályhoz, minden adattagot a paraméterlistából állít be.
   3. Az osztálynak legyen egy *ujEmelet* metódusa, ami nem vár paramétert, és egy új emeletet rak a tortára.
   4. Az osztálynak legyen egy boolean *kremmelMegken* metódusa, ami nem vár paramétert. Ha a torta még nincs megkenve krémmel, úgy a metódus tegye ezt meg. Térjen vissza logikai értékkel attól függően, hogy sikerült-e.
   5. Készíts, egy int típussal visszatérő *mennyiKaloria* metódust az osztályhoz. A torta minden emelete 1000 kalória értékű, ha még krémmel is meg van kenve, akkor ennek a kétszerese.
   6. Hozz létre két tortát. Az egyik tortát kétszer is kend meg krémmel, ennek eredményét mindig írd konzolra. A másik tortára rakj egy emeletet.
10. Írj osztályt, ami egy ***Ember*** objektumot valósít meg.  
    Minden metódus írja ki a konzolra, amit éppen elvégzett, pl. Ha bekapcsolt akkor írja ki, hogy bekapcsolt.
    1. Az ember adattagjai a következőek legyenek: vezetéknév (String), keresztnév (String), születési év (int).
    2. Készíts az osztályhoz konstruktor, ami paraméterek alapján állítja be az adattagokat.
    3. Az osztálynak legyen egy *hogyHivjak* metódusa, ami visszaadja az ember teljes nevét (vezetéknév+keresztnév).
    4. Hozz létre egy ember objektumot, és írd ki a nevét.
11. Írj osztályt, ami egy ***Harcos*** objektumot valósít meg.  
    Minden metódus írja ki a konzolra, amit éppen elvégzett, pl. Ha bekapcsolt akkor írja ki, hogy bekapcsolt.
    1. Az ember adattagjai a következőek legyenek: név (String), életerő (int), harci erő (int).
    2. Készíts az osztályhoz konstruktor, ami paraméterek alapján állítja be az adattagokat.
    3. Az osztálynak legyen egy boolean-al visszatérő *harcol* metódusa, ami egy másik harcost kap paraméternek. A metódus mindkét harcos életerejét csökkentse a másik harcos harci erejével. Ha valamelyik harcos elveszti a harcot (életereje 0 alá csökkenne), a metódus térjen vissza igazzal, egyébként hamissal.
    4. Készítsd el a megfelelő metódusokat az adattagok lekérdezéséhez és módosításához (set, get).
    5. Hozz létre két harcos objektumot. Harcoljanak, amíg valamelyikük el nem veszti a harcot.