1. **feladat**

Készíts egy *Allat* osztályt

* Minden állatnak van éhsége, ami egy egész szám
* Minden állatnak van szomja, ami egy egész szám
* Amikor egy állat létrejön 50-es az éhsége és 50-es a szomja
* Minden állat tud csinálni dolgokat:
  + eszik() az éhsége csökken eggyel
  + iszik() a szomja csökken eggyel
  + jatszik() az éhsége és szomja növekszik eggyel

1. **feladat**

Írj osztályt, ami egy *Macska*objektumot valósít meg.

* A macska adattagjai a következők legyenek:
  + **név (String), súly (double), éhes -e (boolean).**
* Két konstruktort is készíts az osztályhoz. Az egyik általános legyen, ami minden adattagot a konstruktor paraméterlistájából állít be, illetve egy másik, ami az első két adattagot a konstruktor paraméterlistájából kapja, és alapértelmezetten éhes a macska legyen.
* Az osztálynak legyen egy *eszik* metódusa, ami egy double értéket vár (étel mennyisége), és egy 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.
* Az osztálynak legyen egy void futkosmetódusa, ami nem vár paramétert. A macska súlya csökkenjen 0.1-el, és ha nem volt éhes, akkor éhezzen meg.
* ~~Készíts toString metódust az osztályhoz.~~

A mainmetódusban hozz létre két macskát a két különböző konstruktorral, és próbáld meg megetetni őket. Az etetés sikerességéről írj információt a konzolra. Mindkét macska futkosson, utána írd ki szövegesen a konzolra az objektumokat.

1. **feladat**

Írj osztályt, ami egy Emberobjektumot valósít meg.

* Az ember adattagjai a következőek legyenek: vezetéknév (String), keresztnév (String), születési év (int). Az adattagok csak ebből az osztályból legyen elérhetőek.
* Készíts az osztályhoz konstruktor, ami paraméterek alapján állítja be az adattagokat.
* Az osztálynak legyen egy hogyHivjak metódusa, ami visszaadja az ember teljes nevét (vezetéknév+keresztnév).
* • A main metódusban hozz létre egy ember objektumot, és írd ki a nevét.

1. **feladat**

Írj osztályt, ami egy **Szamitogep**objektumot valósít meg.

* A számítógép adattagjai a következőek legyenek:
  + szabad memória MB-ban (double),
  + be van -e kapcsolva ( boolean ).
* Készíts két konstruktort is az osztályhoz. Az egyik általános legyen, ami minden adattagot a paraméterlistából állít be, a másik egy alapértelmezett konstruktor legyen, ami 1024 MB memóriával, kikapcsolva hozza létre a gépet.
* Az osztálynak legyen egy void **kapcsol** metódusa, ami nem vár paramétert. Ha a gép ki van kapcsolva, akkor kapcsolja be, egyébként kapcsolja ki.
* Az osztálynak legyen egy boolean program **Masol** metódusa, ami egy program méretét várja paraméternek MB-ban (double). 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.
* ~~Készíts toString metódust az osztályhoz.~~

*A mainmetódusban hozz létre két számítógépet a fenti konstruktorokkal. Mindkét gép kikapcsolt állapotban kezdjen. Az alapértelmezett gépet kapcsold be, és másold rá először 800 MB, aztán 400 MB programot. A másik gépre másolj 1 MB programot. A másolások eredményeit írd ki.*

**Mindkét objektumot írd ki szövegesen!**

1. **feladat**

Írj osztályt, ami egy **Hallgato**objektumot valósít meg.

* A hallgató adattagjai a következőek legyenek: azonosító (String), évfolyam (int), kreditszám ( nt ).
* Két konstruktort is készíts az osztályhoz. Az egyik általános legyen, ami minden adattagot a konstruktor paraméterlistájából állít be, illetve egy másik, ami az első adattagot a konstruktor paraméterlistájából kapja, évfolyama 1 és kreditszáma 0 legyen.
* Az osztálynak legyen egy void targyFelveszmetódusa, amivel egy int paramétert ( tárgy kreditértéke) kér. A hallgató kreditszáma nőjön a kapott értékkel.
* Az osztálynak legyen egy boolean vizsgazikmetódusa, 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.
* ~~Készíts toString metódust az osztályhoz.~~
* A *main*metódusban hozz létre két hallgatót a két különböző konstruktorral. Az egyikük vegyen fel tárgyat, majd vizsgáztasd őket. A vizsga sikerességéről írj információt konzolra. • Ezután mindkét hallgatót írd ki szövegesen.

1. **feladat**

Írj osztályt, ami egy **Torta**objektumot valósít meg.

* A torta adattagjai a következőek legyenek: emeletek száma (int), meg van -e kenve krémmel (boolean).
* Készíts **két konstruktort** is az osztályhoz. Az egyik általános legyen, ami minden adat- tagot paraméterlistából állít be, a másik egy alapértelmezett konstruktor legyen, ami 1 emeletes, krém nélküli tortát hoz létre.
* Az osztálynak legyen egy **void ujEmelet** metódusa, ami nem vár paramétert, és egy új emeletet rak a tortára.
* 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.
* Készíts egy int típusú **mennyiKaloria** metódust az osztályhoz. A torta minden emelete 1000 kalória értékű, ha még krémmel is meg van kenve, akkor ennek a kétszerese.
* ~~Készíts toString metódust az osztályhoz.~~
* A *main*metódusban hozz létre két tortát a két konstruktorral. Az alapértelmezett tortát kétszer is kend meg krémmel, ennek eredményét mindig írd konzolra. A másik tortára rakj egy emeletet.
* ~~•Mindkét objektumot írd ki szövegesen.~~