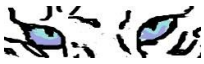
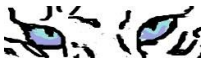


# Virágkötészet

Ebben a feladatban egy virágkötészeti vállalkozás adminisztrációs szoftverének egy részét kell elkészítened. A feladathoz elkészített interfész és az osztályok külön osztály könyvtárba (Class Library) kerüljenek! A feladat elkészítésénél törekedj az **OOP elvek** és a **clean code** szabályok betartására! A feladatleírásban meghatározott osztályokat a feladatok végrehajtása során szükség szerint kiegészítheted, és új osztályokat is létrehozhatsz.

1. Írj egy osztályt **`Alapanyag`** néven, amely egy-egy alapanyaghoz tartozó információkat tartalmaz!
    a. Az osztály kialakításához tanulmányozd az `alapanyagok.txt` állományt. Az elkészítési idő percben értendő.
    b. Az osztály konstruktora kapja meg az alapanyag azonosítóját, nevét, árát és az elkészítési időt.
2. Írj egy osztályt **`Katalogus`** néven, amely az alapanyagokhoz tartozó információkat tartalmazza!
    a. Az osztály konstruktora kapja meg paraméterként az alapanyagok sorozatát.
    b. Az osztályhoz készüljön indexer, amely egy alapanyag azonosítóját megkapva visszaadja az alapanyag adatait.
3. Készíts egy interfészt **`ITermek`** néven, amely 4 csak olvasható tulajdonságot deklarál. A `Tipus` és a `Megnevezes` tulajdonság szöveg értékkel tér vissza, az `ElkeszitesiIdo` és az `Ar` tulajdonság egész típussal tér vissza.
4. Írj egy osztályt **`Termek`** néven, mely implementálja az `ITermek` interfészt!
    a. Az osztály kialakításához tanulmányozd a `termekek.txt` állományt.
    b. Az osztály konstruktora kapja meg paraméterként a termék azonosítóját, típusát, megnevezését, a termék előállításához szükséges alapanyagokat és hogy melyik alapanyagból mennyi kell a termék elkészítéséhez, valamint az alapanyagkatalógust.
    c. Az `ElkeszitesiIdo` tulajdonság visszaadott értéke az egyes alapanyagok elkészítési idejének összegével tér vissza. Ha valamelyik alapanyagból több kell, akkor az alapanyag elkészítési ideje annyiszor számít, ahány darab kell belőle.
    d. Az `Ar` tulajdonság visszaadott értéke az alapanyagonkénti árak összege.
    e. Az osztálynak legyen `ToString` metódusa.
5. Készíts egy osztályt **`Termekek`** néven, mely az előállítható termékek tárolására alkalmas.
    a. Az osztálynak legyen indexere, amely egy termék azonosítóját megkapva visszaadja a termék adatait.
    b. Az osztálynak legyen `ToString` metódusa.
6. Készíts egy osztályt **`FeladatLista`** néven, amely egy dolgozó feladatainak kezelésére alkalmas.
    a. A feladatlista kezdetben legyen üres.
    b. Készítsd el az osztályhoz a + operátort, amely úgy működik, hogy az eddigi feladatok listáját kiegészíti egy új elkészítendő termékkel, és egy új feladatlistát ad vissza.
    c. Az aktuális feladatlista legyen példány szinten lekérdezhető.
7. Készíts egy osztályt **`Dolgozo`** néven, amelyben egy-egy munkavállaló adatait tárolod.
    a. Az osztály kialakításához tanulmányozd a `dolgozok.txt` állományt.
    b. Az osztály konstruktora kapja meg paraméterül a dolgozó azonosítóját és a nevét.
    c. Az osztálynak legyen egy `Gyakorlottsag` tulajdonsága, amely csak lekérdezhető, valós típussal tér vissza, és nem tartalmaz megvalósítást.
    d. Az osztálynak legyen egy `MunkaraForditottIdo` tulajdonsága, amely csak lekérdezhető, egész típussal tér vissza, és nem tartalmaz megvalósítást.



   e. Az osztálynak legyen egy `UjFeladatHozzaadasa` metódusa, amely egy terméket kap paraméterül, és a terméket hozzáadja a dolgozó feladatlistájához.
   f. Az osztálynak legyen `ToString` metódusa, amely megadja a dolgozó nevét és a munkára fordított idejét percben.
8. Készíts egy osztályt **`Viragkoto`** néven, amely a `Dolgozo` osztályból származik.
   a. A virágkötőknek a `Gyakorlottsag` tulajdonsága 100.
   b. Az osztályban a MunkaraForditottIdo tulajdonság értékét úgy számoljuk ki, hogy a feladatlistában található termékek elkészítési idejének összege.
9. Készítsd el a **`HibasFeladatException`** osztályt, amely az `Exception` osztályból származik, és „*`A feladathoz nincs elegendő tudása a gyakornoknak.`*” szöveggel dob hibaüzenetet!
10. Készíts egy osztályt **`Gyakornok`** néven, amely a `Dolgozo` osztályból származik.
   a. A gyakornok konstruktora kapja meg paraméterül azoknak a termékeknek az azonosítóját sorozat típusként, amit a gyakornok már el tud készíteni.
   b. A gyakornoknak a `Gyakorlottsag` tulajdonsága véletlenszerűen 70, 80 vagy 90.
   c. Az osztályban a MunkaraForditottIdo tulajdonság értékét úgy számoljuk ki, hogy a feladatlistában található termékek elkészítési idejét megnöveljük annyiszorosával, ahány százalék hiányzik a `Gyakorlottsag` értékből a 100%-hoz viszonyítva, és így összegezzük az előállítási időket.
   d. Az osztályban az `UjFeladatHozzaadasa` metódus dobjon kivételt, ha a gyakornok nem tudja elkészíteni a kapott feladatot, különben pedig adja hozzá a terméket a gyakornok feladatlistájához.
   e. Az osztálynak legyen `ToString` metódusa, amely az ősosztályban megadott adatokon kívül azt is kiírja zárójelben, hogy gyakornokról van szó.
11. Készíts egy osztályt **`MunkaeroFelvetel`** néven, amellyel a dolgozók létrehozását kezeled.
12. Készíts egy osztályt **`Dolgozok`** néven, amellyel a felvett dolgozókat tárolod.
   a. Az osztálynak legyen indexere, amely egy dolgozó azonosítóját megkapva visszaadja a dolgozó adatait.
   b. Az osztály adja vissza egy tulajdonságban a tárolt dolgozók számát.
13. Készíts egy osztályt **`FeladatKiosztas`** néven, amellyel a dolgozókhoz hozzárendeled a hozzájuk tartozó feladatokat.
   a. Amennyiben a dolgozó nem tudja ellátni a kiosztott feladatot, akkor a hibalista.txt állományba kerüljön be erről egy hibaüzenet, ami a hibaüzeneten kívül tartalmazza a dolgozó id-t, valamint a termék id-t is.
14. A futtatható programban a következő feladatokat kell megvalósítani az elkészített osztályok használatával:
   a. Az alkalmazás olvassa be és dolgozza fel a kapott forrásállományokat.
   b. Jelenítsd meg, hogy milyen termékeket lehet elkészíteni!
   c. Jelenítsd meg, hogy melyik dolgozó hány percet dolgozott a rá osztott feladatokkal összesen!