

Mikulás játékgyára

Az ünnepek közeledtével Mikulás játékgyarában a korábbiaknál is szorgalmasabban dolgoznak a manók, hogy a jó gyerekeknek szánt összes ajándék időben elkészüljön. Ebben a feladatban a játékgyár adminisztrációs szoftverének egy részét kell elkészítened, ezzel is segítve a manók munkáját.

Készíts egy mappát a saját neveddel a számítógéped asztalán! A feladat megoldását ebben a mappában készítsd el!

A feladat megoldásához használhatod a <https://learn.microsoft.com/en-us/dotnet/csharp/> hivatalos dokumentációt!

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 létrehozatsz. Az eredmények megjelenítésénél kövesd a megadott mintát!

1. Készíts egy osztályt **GyartasAdat** néven, amely egy-egy játékhoz tartozó gyártási információkat tartalmazza!
 - a. Az osztály kialakításához tanulmányozd a `gyartas.txt` állományt. Az elkészítési idő percben értendő.
 - b. Az osztály konstruktora kapja meg a gyártási folyamat azonosítóját, típusát és az elkészítési időt.
2. Készíts egy osztályt **GyartasAdatok** néven, amely az összes játékhoz tartozó gyártási információkat tartalmazza!
 - a. Az osztály konstruktora kapja meg paraméterként a gyártási információk sorozatát.
 - b. Az osztályhoz készüljön indexer, amely egy gyártás azonosítóját megkapva visszaadja a gyártás adatait.
 - c. Az osztályhoz készüljön `ElerhetoGyartasAzonosito` néven egy tulajdonság, amely visszaadja névsorrendben a gyártási azonosítókat.

(Javaslat: Nézd meg a 12. feladat a. és b. pontját, a feladat az eddigi osztályok elkészítése után elvégezhető, így tudod ellenőrizni a munkádat.)

3. Készíts egy interfészt **IJatek** néven, amely 4 csak olvasható tulajdonságot deklarál. Az `Azonosito`, a `Tipus` és a `Megnevezes` tulajdonság szöveg értékkel tér vissza, az `ElkeszitesiIdo` tulajdonság egész típussal tér vissza.
4. Írj egy osztályt **Jatek** néven, mely implementálja az `IJatek` interfészt!
 - a. Az osztály kialakításához tanulmányozd az `ajandekok.txt` állományt. (A modulok azonosítói csak interaktív játékoknál vannak megadva, egyszerű játékoknál nincsenek.)
 - b. Az osztály konstruktora kapja meg paraméterként a játék azonosítóját, típusát, megnevezését, valamint a gyártási adatokat.
 - c. Az `ElkeszitesiIdo` tulajdonság nem tartalmaz megvalósítást.
 - d. Az osztály `ToString` metódusa a játék megnevezését adja vissza.
5. Készíts egy osztályt **EgyszeruJatek** néven, amely a `Jatek` osztályból származik, és belőle nem származtatható új osztály.
 - a. Az `ElkeszitesiIdo` tulajdonság adja vissza a gyártási adatok indexerének felhasználásával a játék elkészítési idejét!

6. Készíts egy osztályt **InteraktivJatek** néven, amely a **Jatek** osztályból származik, és belőle nem származtatható új osztály.
 - a. Az osztály konstruktora kapja meg paraméterül azoknak a moduloknak az azonosítóját is sorozat típusként, amilyen modulokból elkészíthető a játék.
 - b. Készíts egy szöveg típussal visszatérő metódust **TípusMeghatározas** néven, amely paraméterül kap egy modul azonosítót szöveggént, és visszaadja belőle a típus azonosítót. Amennyiben a modul azonosítójának első karaktere 'k' betű, akkor a típus azonosító megegyezik a modul azonosítóval, ha nem 'k' betű, akkor a modul azonosító első betűje lesz a típus azonosító.
 - c. Az **ElkészitesiIdo** tulajdonság adja vissza a gyártási adatok indexerének felhasználásával a játék elkészítési idejét úgy, hogy a modulok gyártási idejét összeadja.
7. Készíts egy osztályt **JatekFactory** néven, amely a játék osztályok példányainak előállítására szolgál!
 - a. Az osztály kialakításához tanulmányozd az **ajandekok.txt** állományt.
 - b. Az osztály **Factory** metódusa kapja paraméterül az állomány egyik sorát, valamint a gyártási adatokat! Amennyiben az ajándék típusa „i”, akkor interaktív játékot kell készíteni, minden más esetben pedig egyszerű játékot.
8. Készíts egy osztályt **Jatekok** néven, amely az összes elkészíthető játék adatait tartalmazza!
 - a. Az osztály konstruktora kapja paraméterül sorozat típusként az elkészíthető játékokat.
 - b. Az osztálynak legyen indexere, amely egy játék azonosítóját megkapva visszaadja a játék adatait.
 - c. Készíts egy tulajdonságot **JatekTipusok** néven, amely megnevezés szerint névsorba rendezve visszaadja az elkészíthető játékokat, kihagyva belőle azokat, amelyeknek típusa 'k' betűvel kezdődik!

(Javaslat: Nézd meg a 12. feladat c. és d. pontját, a feladat az eddigi osztályok elkészítése után elvégezhető, így tudod ellenőrizni a munkádat.)

9. Készítsd el a **TulSokFeladatException** osztályt, amely az **Exception** osztályból származik, és „Túl sok feladat, több mint 8 óra elkészíteni.” szöveggel dob hibaüzenetet!
10. Készíts egy osztályt **Feladat** néven, amely 1-1 elvégzendő feladat adatait fogja tárolni. Mivel egészségvédelmi okokból a manók nem dolgozhatnak többet egyszerre 8 óránál, ezért nem hozható létre olyan feladat, aminek elvégzése 8 óránál tovább tart.
 - a. Az osztály kialakításához tanulmányozd a **feladatok.txt** állományt.
 - b. Az osztályban tárold adatként, hogy maximum hány perc állhat rendelkezésre a feladatok elkészítésére!
 - c. Az osztály konstruktora kapja paraméterül az elkészítendő játékot, valamint az elkészítendő darabszámot.
 - d. Készíts egy tulajdonságot **ElkészitesiIdo** néven, amely visszaadja a feladat elkészítési idejét percben úgy, hogy a játéktípus elkészítési idejét megszorozza a darabszámmal.
 - e. Amennyiben a feladat elkészítési ideje több, mint 8 óra, akkor a feladat ne jöjjön létre, helyette válts ki **TulSokFeladatException** kivételt!
 - f. Az osztály **ToString** metódusa adja vissza a játék megnevezését, az elkészítendő darabszámot, valamint az elkészítési időt percben. A metódus elkészítéséhez tanulmányozd a mintát!

11. Készíts egy osztályt **Feladatok** néven, amely az elkészítendő összes feladatot fogja tárolni, de csak azokat, amik 8 órán belül elvégezhetők.
- Készítsd el az osztályhoz a + operátort, amely úgy működik, hogy az eddigi feladatok listáját egészítse ki egy új feladattal, és az új feladatlistával térjen vissza.
 - Készíts egy tulajdonságot `FeladatLista` néven, amely sorozat típusként visszaadja a feladatlistában szereplő adatokat.

(Javaslat: Nézd meg a 12. feladat e. és f. pontját, a feladat az eddigi osztályok elkészítése után elvégezhető, így tudod ellenőrizni a munkádat.)

12. A futtatható programban a következő feladatokat kell megvalósítani az elkészített osztályok használatával:
- Olvasd be és dolgozd fel a `gyartas.txt` forrásállományt.
 - Jelenítsd meg a minta szerint az elérhető gyártás azonosítókat!
 - Olvasd be és dolgozd fel az `ajandekok.txt` forrásállományt.
 - Jelenítsd meg a minta szerint a gyártott játékokat!
 - Olvasd be a `feladatok.txt` forrásállományt.
 - Amennyiben a megadott játéktípus nem szerepel az elérhető játéktípusok között, akkor készüljön erről egy hibaüzenet a `hibalista.txt` állományba, a megadott minta szerint!
 - Amennyiben a feladat létrehozása kivételt váltana ki, akkor készüljön erről egy hibaüzenet a `hibalista.txt` állományba, a megadott minta szerint!
 - A feladatokat egyenként add hozzá a feladatok listájához a + operátor használatával!
 - Gondoskodj arról, hogy a `hibalista.txt` állomány a program minden indításakor üres legyen, ne maradjanak benne az előző futtatások hibái!
 - Jelenítsd meg a minta szerint az elvégezhető feladatokat!
 - Jelenítsd meg a minta szerint játékonként, hogy összesen hány darabot kell készíteni belőlük a tárolt feladatlista alapján!

A feladathoz minták a következő oldalon találhatóak.

Minta:

Elérhető gyártás azonosítók: b; j; k1; k2; p

A gyártott játékok: hajás baba, interaktív plüss cica, interaktív plüss kutya, interaktív plüss tigris, irányítható vonat, játék autó, mozdony vagonokkal, plüss cica, plüss kutya, plüss macskó, plüss oroszlán, plüss párduc, plüss pingvin, plüss róka, plüss tigris, puha testű játék baba, távirányítós játék autó

Az elvégzendő feladatok:

plüss tigris: 12 db, elkészítési idő: 300 perc
plüss cica: 8 db, elkészítési idő: 200 perc
plüss párduc: 10 db, elkészítési idő: 250 perc
plüss róka: 7 db, elkészítési idő: 175 perc
interaktív plüss kutya: 6 db, elkészítési idő: 300 perc
interaktív plüss tigris: 6 db, elkészítési idő: 300 perc
mozdony vagonokkal: 15 db, elkészítési idő: 375 perc
irányítható vonat: 10 db, elkészítési idő: 400 perc
interaktív plüss kutya: 5 db, elkészítési idő: 250 perc
plüss tigris: 15 db, elkészítési idő: 375 perc
interaktív plüss tigris: 8 db, elkészítési idő: 400 perc
plüss oroszlán: 9 db, elkészítési idő: 225 perc
plüss róka: 5 db, elkészítési idő: 125 perc
interaktív plüss kutya: 6 db, elkészítési idő: 300 perc

Játékonként a készítendő darabszámok:

plüss tigris: 27 db
plüss cica: 8 db
plüss párduc: 10 db
plüss róka: 12 db
interaktív plüss kutya: 17 db
interaktív plüss tigris: 14 db
mozdony vagonokkal: 15 db
irányítható vonat: 10 db
plüss oroszlán: 9 db

Minta a hibalista.txt állomány tartalmára:

Túl sok feladat, több mint 8 óra elkészíteni. - 12 db interaktív plüss cica: 600 perc
Túl sok feladat, több mint 8 óra elkészíteni. - 12 db interaktív plüss tigris: 600 perc
Túl sok feladat, több mint 8 óra elkészíteni. - 20 db játék autó: 500 perc
Túl sok feladat, több mint 8 óra elkészíteni. - 30 db hajás baba: 600 perc
x: Ilyen azonosítójú játékot nem gyártanak.
Túl sok feladat, több mint 8 óra elkészíteni. - 600 db plüss tigris: 15000 perc