

Feladat – Nagyzh – FONTOS INFÓK

- A feladat során alkalmazd a megtanult objektum-orientáltsági elveket!
- A fájlokat egy zip-be csomagolva kell feltölteni.
- Az alapkódban a Main függvény nem módosítható! A kód többi részében, ahol a feladat szerint szükséges, végezhetőek apróbb módosítások, de a programnak működőképesnek kell lennie!
- A feladatok egymásra épülhetnek. Az 1-3. feladatpontig a feladatokat mindenképp meg kell oldani, anélkül nem fognak működni a továbbiak.

A feladat leírása

A feladat egy kisbaba termékekkel foglalkozó áruház árukezelésének megvalósítása. Az áruházban különböző kisbaba termékek (**BabyProduct**) érhetőek el, a programnak ezek nyilvántartását kell tudnia kezelnie. Jelenleg háromfajta termékkörrel foglalkoznak, úgymint játékok, ruházat, etetőszékek. A kisbaba termékek esetén nyilvántartjuk az azonosítóját (szöveg), a nevét (szöveg), az árát (forintban, egész) és a készletmennyiséget (darab, egész). A játékok esetében ezen kívül még ismerjük, hogy hány éves gyerekeknek szánták azt (egész). A ruházatnál a ruha méretét ismerjük (szöveg). Az etetőszékek esetében még azt tudjuk, hogy a szék magassága állítható-e (bool). TIPP: ABSZTRAKT ŐSOSZTÁLY + GYEREKOSZTÁLYOK.

A gyerekosztályokban legyen felüldefiniálva a *ToString* metódus, amely az adott termék tulajdonságait adja vissza a példakimenetben látható módon. Minden osztálynak legyen 2 konstruktora: az egyik paraméter nélküli, a másik paraméterben várja a tulajdonságok értékeit és beállítja azokat. A paraméteres konstruktorok a következő paraméterezéssel legyenek ellátva: (azonosító, név, ár, készletmennyiség, gyerekosztálynak megfelelő plusz tulajdonság).

A működést az **IStock** interfészt megvalósító osztályon keresztül kell megoldani. Az osztálynak legyen egy paraméter nélküli konstruktora, amely példányosítja a tárolót, amelyben a termékeket tároljuk. Legyen továbbá egy generikus *Load* metódusa, amely egy JSON fájl nevét kapja meg és onnan betölti a megadott típus szerinti árukészletet. Feltehetjük, hogy helyesek az adatok, és minden termékhez van pontosan egy bejegyzés a raktár készletben. TIPP: WHERE T : ŐSOSZTÁLY

Az osztálynak van egy *List* metódusa, amely a teljes terméklistát adja vissza, a List metódusból legyen még egy, ami egy generikus metódus, és csak a megadott típusnak megfelelő termékek listáját adja vissza. Ezenkívül van egy *Get* metódusa, amely pedig azonosító szerint ad vissza egy terméket. Az *Add* metódus egy új terméket ad hozzá a készlethez, a *Remove* metódus pedig azonosító alapján eltávolítja a terméket a készletből. TIPP: IF(PRODUCT IS TPRODUCT) VAGY LIST.OFTYPE<TPRODUCT>().TOLIST()

A projekthez tartozik egy másik fajta Json fájl is, ami egy vásárló által leadott rendelés adatait írja le. Ebben egy lista van, amelynek minden eleme egy termék azonosítót és egy darabszámot tárol. A **Program** osztályban legyen egy *LoadOrder* metódus, amely megkapja egy ilyen fájl nevét, beolvassa a

tartalmát, és visszaad egy belőle feltöltött **Order** objektumot. TIPP: EDIT -> PASTE SPECIAL -> PASTE JSON AS CLASSES.

Az **IStock** interfészt megvalósító osztálynak legyen egy **Payment** metódusa, amely megkap paraméterben egy rendelés objektumot, kiszámolja és visszaadja a rendelésben szereplő termékek alapján a rendelés végösszegét. A készletünkben az áruk árait nettó értékben tartjuk nyilván, a végösszegnek viszont az áfával növelt összegnek kell lennie (27%-os áfakulcs esetén), az összeg egész forint legyen. Ezen kívül még fel kell számolni egy egyszeri kényelmi díjat is, ami bruttó 320 forint egy megrendelésre.

Legyen egy **Sale** metódusa is az **IStock** interfészt megvalósító osztálynak, amely a raktár készletből levonja a rendelt mennyiségeket. Amennyiben bármelyik termékből nincs elég a raktáron, a teljes rendelés legyen érvénytelen. A rendelés érvényességét a visszatérési érték jelezze. TIPP: ELLENŐRIZNÉM ELŐBB A TELJESÍTHETŐSÉGET, ÉS UTÁNA VONNÁM LE A MENNYISÉGEKET.

Az **IStock** interfészt megvalósító osztálynak legyen egy **Upload** metódusa, amely egy termék azonosítót kap és egy darabszámot, és az adott termékből elérhető mennyiséget megfelelően növeli (ha nem létezik a termék, akkor nem csinál semmit).

Az **IStock** interfészt megvalósító osztálynak legyen egy generikus **Save** metódusa, amely a megadott típusú termékeket menti ki a paraméterben megadott fájlba. TIPP: WHERE T : ŐSOSZTÁLY

Pontozás

- | | |
|--|----------|
| 1. Termékek adatszerkezete | (6 pont) |
| 2. IStock interfész megvalósítása, Load metódus | (4 pont) |
| 3. List és Get metódusok | (4 pont) |
| 4. Add és Remove metódusok | (4 pont) |
| 5. Rendelés adatszerkezete (Order osztály) és betöltése (LoadOrder) metódus | (6 pont) |
| 6. Payment metódus | (4 pont) |
| 7. Sale metódus | (4 pont) |
| 8. Upload metódus | (4 pont) |
| 9. Save metódus | (4 pont) |