Feladat – Nagyzh – FONTOS INFÓK

- A feladat során alkalmazd a megtanult objektum-orientáltsági elveket!
- A fájlokat egy zip-be csomagolva kell feltölteni.

A feladat leírása

A feladatban vásárlások során alkalmazható kuponok kezelése a cél. Minden kuponnak van egy értéke, hogy mennyi forintot vesz le a vásárlásból, és implementálnia kell az **IUsable** interfészt. Utóbbi egy használható objektumot jelöl, és kettő metódust tartalmaz: *IsActive*, ami visszaadja, hogy az adott objektum aktív-e (kupon esetén érvényes-e), valamint *Use*, ami elhasználja az objektumot és visszatér a sikerességgel (ha érvénytelen, akkor nem tudja használni, és **false** a visszatérés). Az ősosztály neve **Coupon** legyen (a *Main*-hez kell).

A kuponoknak több fajtájuk is van. Az egyszeri kupon csak egyszer használható, utána érvénytelen lesz. A többszöri kuponhoz tartozik egy mennyiség, hogy hány alkalommal használható, és majd ennyi használat után lesz érvénytelen. A havi kupon egy adott hónap során akárhányszor használható, azonkívül nem. A havi kuponnak tudnia kell, hogy melyik hónapra érvényes, és ennek vizsgálatakor az aktuális dátummal hasonlítja össze. Az aktuális dátumot a programban egy állítható statikus adat tárolja (alapértelmezetten legyen 2024.05.01).

A *Main* függvényben a jelölt helyen töltsd fel a listát egy egyszeri kuponnal (1500 Ft értékű), egy 5-ször használható többszöri kuponnal (500 Ft), valamint egy havi kuponnal, ami 2024 májusában érvényes (200 Ft).

A kuponokat szeretnénk tényleges vásárlás során is alkalmazni. Legyen egy **Purchase** osztály, ami egy teljes vásárlás adatait tárolja. Létrehozáskor a konstruktor egy JSON fájl nevét várja, amiből töltse is be az adatokat. A fájlban egy JSON tömb van, benne minden adat egy termék: név, egységár, mennyiség (ami tovább bomlik számszerű mennyiségre és mértékegységre, utóbbi csak a megjelenítés miatt fontos). TIPP: EDIT -> PASTE SPECIAL -> PASTE JSON AS CLASSES.

A **Purchase** osztálynak legyen egy *Print* metódusa, amely megjeleníti a vásárolt termékek adatait. Legyen egy *TotalCost* metódusa, amely kiszámítja és visszaadja a vásárlás költségét. Legyen egy *CostWithCoupons* metódusa, amely szintén meghatározza és visszaadja a teljes költséget, de paraméterként egy kupon listát vár, és minden használható kupont használ és levonja a kedvezményüket. Feltehetjük, hogy nem megy negatívba az érték.

A **Program** osztályban legyen egy *LoadCoupons* metódus, amely paraméterben egy JSON fájl nevét kapja, betölti, és visszaadja listaként. A fájl egy tömbben kuponokat tárol (bármilyen fajtát) a főbb adatokkal, és a fájlban minden kupon új (még nem volt használva).

Legyen egy generikus *CountCoupons* metódus is, amely a generikus típusnak megfelelő kuponok számát adja vissza a paraméterben kapott listából. Egy logikai paramétere is legyen, ami, ha igaz, akkor csak az érvényes kuponokat számolja, ha hamis, akkor mindet. A *Main*-ben az ide tartozó tesztkódban hiányoznak a generikus paraméterek, azt is töltsd ki a megfelelő típusokkal. TIPP: WHERE T : ŐSOSZTÁLY ; IF (… IS …)

Legyen még egy *ExportCounts* metódus a **Program** osztályban, amely szintén egy kupon listát kap, valamint egy JSON fájlnevet, és a fájlba kimenti, hogy az egyes kupon típusokból mennyi érvényes és érvénytelen van a listában.

Pontozás

1. Kuponok adatszerkezete **(4 pont)**
2. Interfész megvalósítása, metódusok helyes működése **(4 pont)**
3. Kuponok létrehozása a *Main*-ben **(2 pont)**
4. Vásárlás adatszerkezete (**Purchase** osztály) és betöltése **(6 pont)**
5. *Print* metódus **(4 pont)**
6. *TotalCost* metódus **(4 pont)**
7. *CostWithCoupons* metódus **(4 pont)**
8. *LoadCoupons* metódus **(4 pont)**
9. *CountCoupons* metódus **(4 pont)**
10. *ExportCounts* metódus **(4 pont)**