

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 egy új gyorsétterem kínálatát kezeljük. Három fajta ételt készítenek: pizzát, hamburgert és kebabot. Minden ételnek van egy neve (szöveg), ami alapján hivatkozhatunk rá. Ezen felül a pizzához tároljuk a feltéteket (szövegek), a hamburgerhez a húspogácsa típusát (marha, csirke, vegetáriánus) és méretét (grammban, egész), valamint, hogy van-e benne bacon vagy sajt (logikai értékek), kebabhoz pedig a szósz típusát (édes, csípős, fokhagymás), valamint, hogy van-e benne hagyma, paradicsom, illetve feta sajt (logikai értékek). Az **ősosztály** neve **Food** legyen, a pizzához pedig a **Pizza** osztály tartozzon (későbbi feladathoz fog kelleni).

Az étterem étlapját a **FoodBar** osztályon keresztül fogjuk kezelni, ami tárolja az összes ételt, amit el tudnak készíteni. Az étlapot egy JSON fájlban tároljuk, aminek nevét az osztály konstruktora kapja meg, és végezze is el a betöltést. A fájlban minden ételnél szerepel a típusa szöveggént, majd a típusának megfelelő adatok.

A **FoodBar** osztályban legyen egy **printAll** metódus, amely kilistázza az étlapot a minta szerint. TIPP: CÉLSZERŰ HOZZÁ VIRTUÁLIS FÜGGVÉNY

Az étlapon jelenjen meg az egyes ételek ára is. Pizza esetében az alapár 2000 Ft, és minden feltét még extra 300 Ft-ot tesz erre rá. Hamburger esetében 500 Ft az alapár, a húspogácsa ára marhahús esetén 10 Ft/g, csirkehús esetén 6 Ft/g, vegetáriánus verzió esetén 7 Ft/g, a bacon és a sajt (ha vannak) 100-100 Ft-ot tesznek rá. Kebab esetén az alapár 1800 Ft, ezen felül (amennyiben van) a hagyma +150 Ft, a paradicsom +100 Ft, a feta sajt pedig +250 Ft, a szósz nem befolyásolja az árat. TIPP: VIRTUÁLIS METÓDUS

Legyen egy **Order** osztály, ami egy rendelést valósít meg. Az osztályon belül legyen egy **loadOrder** statikus metódus, amely megkapja egy JSON fájl nevét, betölti a rendelést, és visszaadja. A fájlban a rendelés egy tömbként szerepel, minden eleme tárolja egy étel nevét (szöveg), típusát (szöveg), valamint a rendelt mennyiséget (egész). Egyedi összeállítás nincs, csak egy-az-egyben az étlapon szereplő lehetőségek közül lehet választani.

A **FoodBar** osztályban legyen egy **processOrder** metódus, amely megkap egy **Order** objektumot, kilistázza az összes rendelt érték részleteit (a minta szerint), majd megjeleníti a rendelés összes árát is.

A pizzák esetében bizonyos feltétek prémiumnak számítanak, így drágábbak a többinél. Az osztály statikus formában tárolja a prémium feltéteket (több is lehet). Legyen egy **addPremiumTopping** metódus, amely paraméterben egy feltétet kap (szöveg), és elmenti, hogy az prémiumnak számít,

valamint egy `clearPremiumToppings`, ami kiüríti a prémium feltétek vektorát. A pizzák ár számítását módosítani kell: a prémium feltétek 300Ft helyett 500Ft-ba kerülnek.

Az `Order` osztályhoz legyen egy `<<` operátor, mellyel a tartalma kiírható tetszőleges kimeneti folyamra a mintának megfelelően. Ennél a kiírásnál nincs szükség sem az ételek részletes adataira, sem a költségszámításra.

Pontozás

- | | |
|---|----------|
| 1. Ételek adatszerkezete | (4 pont) |
| 2. <code>FoodBar</code> osztály, konstruktor, JSON betöltés | (8 pont) |
| 3. <code>printAll</code> metódus | (3 pont) |
| 4. <code>PrintAll</code> metódus az árakkal együtt | (5 pont) |
| 5. <code>Order</code> osztály, <code>loadOrder</code> metódus | (5 pont) |
| 6. <code>processOrder</code> metódus | (5 pont) |
| 7. Prémium feltétek kezelése | (4 pont) |
| 8. Prémium feltétek beleszámítása az árba | (2 pont) |
| 9. <code><<</code> operátor a rendelésekhez | (4 pont) |