

Programozás II (C#) ismeretek előzetes ellenőrzése - MINTA

- A feladat során alkalmazd a megtanult objektum-orientáltsági elveket!
- A fájlokat egy zip-be csomagolva kell feltölteni.
- A Program.cs elején lévő #define-ok segítségével lehet a teszt kódokat bekapcsolni. Ebben a fájlban mást nem szabad módosítani, kivéve, ha a feladat azt mondja.

A feladat leírása

A feladatban épületek nyilvántartását kezeljük. Minden épületnek van egyedi azonosítója (szöveg), címe (szöveg), alapterülete (egész, m²), és építésének az éve (egész). Három fajta épületet kezelünk. Az egyszerű épület extra adatot nem tárol. A több szintes épület még a szintek (emeletek) számát is tárolja (egész). A harmadik típus a teljes épület, amely még tárolja az épülethez tartozó telek méretét (egész, m²). Az őszosztály neve **Building** legyen.

Az épületeket a Földhivatalt jelző **LandOffice** osztály tárolja. Az osztály rendelkezzen generikus *LoadBuildings* metódussal, amely paraméterben egy fájlnevet kap, és onnan betölti a generikus paraméternek megfelelő típusú épületeket. A *Main*-ben megjelölt helyen hívd meg ezt a metódust háromszor, a **simplebuildings.json**, **multilevelbuildings.json** és **buildingswithland.json** fájlokra (ezek rendre az egyszerű, többszintes, illetve teljes épületeket tárolják). TIPP: WHERE T : ŐSZOSZTÁLY

Az épület osztályok írják felül a *ToString* metódust a mintának megfelelően. A **LandOffice** osztályban legyen egy *ListAll* metódus, amely megjeleníti az összes épület adatait.

Oldd meg, hogy két épület összehasonlítható legyen (rendezés céljából), és az összehasonlítás alapja a cím legyen. A **LandOffice** osztályban legyen egy *SortBuildings* metódus, amely sorba rendezi az épületeket cím szerint. TIPP: MELYIK INTERFÉSZ VALÓ ERRE?

Az épületeknek legyen egy metódusa, amely visszaadja a teljes területet. Egyszerű épület esetén ez a tárolt érték. Több szintes épület esetén a tárolt érték egy szint alapterületét jelzi, de a teljes területhez ezt be kell szorozni a szintek számával. Teljes épület esetén a teljes terület az épület és a telek együttes területe. TIPP: ABSZTRAKT METÓDUS

A **LandOffice** osztályban legyen egy *ListAreas* metódus, amely minden épületnek megjeleníti a címét és a teljes területét (az előző bekezdés metódusát alkalmazva). Legyen tovább egy *ExportAreas* metódus, mely paraméterben egy fájlnevet vár, és abba a fájlba valamilyen JSON formában kiment minden épület azonosítóját, típusát és teljes területét. TIPP: IF (... IS ...) VAGY VIRTUÁLIS METÓDUS

Az épületek esetén szeretnénk tárolni azt is, hogy melyik évben volt felújítva. Egy épület esetén ez többször is megtörténhetett. Ennek megfelelően egészítsd ki az adatszerkezetet, és legyen benne lehetőség új felújítási év hozzáadására. A **LandOffice** osztályban legyen egy *LoadRenovations* metódus, amely egy JSON fájl nevét kapja, és az adott fájlból betölti a felújítások évszámait. A fájlban egy lista van,

melyben minden felújítás egy épület azonosítóját és a felújítás évét tárolja. Ha egy épületből több felújítás is volt, akkor többször is szerepel a listában más évszámmal. Feltehetjük, hogy évszám szerint rendezve vannak az adatok, de nem kell ezzel foglalkozni. Az épületek *ToString* metódusait módosítsd úgy, hogy amennyiben az épülethez volt felújítás, azokat is jelenítse meg.

Pontozás

1. Épületek adatszerkezete **(4 pont)**
2. **LandOffice** osztály, *LoadBuildings* metódus **(8 pont)**
3. *ToString* metódusok **(4 pont)**
4. *ListAll* metódus **(2 pont)**
5. Épületek rendezhetősége **(3 pont)**
6. *SortBuildings* metódus **(2 pont)**
7. Épületek teljes területe **(3 pont)**
8. *ListAreas* metódus **(2 pont)**
9. *ExportAreas* metódus **(4 pont)**
10. Felújítások éveinek tárolása **(2 pont)**
11. *LoadRenovations* metódus **(4 pont)**
12. *ToString* metódus módosítása **(2 pont)**

Értékelés

36-40 pont: 5-ös

31-35 pont: 4-es