

Clean Code, OOP elvek értékelési szempontok

- A létrehozott interfész és az osztályok külön **Class Library**-be kerültek. Az elkészített interfész és az osztályok megfelelő néven készülnek el, külön fájlokban. Legalább még egy projektet tartalmaz a solution.
- Az osztályhoz tartozó **egyszerű adatok** tárolásához **automatikus tulajdonságokat** használ, amennyiben a feladat ezt lehetővé teszi (nincs szükség beállítás védelemre).
- Az osztályokban létrehozott **tulajdonságokban** minimális láthatóságot használ, ha nincs szükség a beállíthatóság bővítésére, akkor **init** résszel rendelkeznek. (Kivéve a számított tulajdonságok.)
- Minden létrehozott **mező private** láthatóságú, amennyiben csak a konstruktor módosítja a tartalmát, akkor **readonly** is.
- A mezőnevek kisbetűvel kezdődnek, a tulajdonságok és a metódusok nagybetűvel, Camel Case írásmódban. Beszédes elnevezéseket használ az osztályok alkotóelemeinél.
- Az osztályokban létrehozott metódusok csak 1 feladatot látnak el, tartalmuk kifer 1 képernyőre, a kódsor olvasásához nem kell jobbra-balra lapozni.
- Az osztályokban létrehozott metódusok megfelelő típusú visszaadott értékkel vagy paraméterekkel kommunikálnak. (Pl.: Ha egy metódusnak 1 szám értéket kell visszaadnia, akkor azt számként adja vissza, és nem a minta alapján kiírandó szöveggént.)
- Konzol és fájl műveletek csak a Program.cs-ben találhatóak.
- A Program.cs-ben létrehozott változók beszédes nevűek, kisbetűvel kezdődnek.
- Minden létrehozott projektje .NET 8.0-s, a konzol alkalmazásnál top level statements sablont használ.
- A teljes alkalmazás szintaktikailag hibátlan, lefordítható.