

Feladat – Kisz 1 – FONTOS INFÓK

- A feladat során alkalmazd a megtanult objektum-orientáltsági elveket!
- A megadott példakódon ne módosíts, hacsak a feladat nem kéri! Ez alól kivétel a Program.cs elején lévő kommentek (#define). A megoldásnak ezen fájlokkal kell mennie, hiszen az ellenőrzés során a Moodle biztosítja őket.
- Figyelj a kiírás megfelelő formátumára, szóközökre, új sorokra! Az automata javító csak a tényleges kimenetet látja, nem tudja mit akartál.
- Minden pont értékeléséhez szükséges, hogy az adott ponthoz tartozó #define szerepeljen a Program.cs fájl elején (#define PART1 az 1. feladathoz, #define PART4 a 4. feladathoz, stb.).
- Csak olyan kódot tölts fel moodle-be, ami nálad fordul. Ami nálad nem fordul, a Moodle-ben sem fog.
- A fájlokat nem tömörítve kell feltölteni, hanem önmagukban (drag-and-drop-pal egyszerre be lehet húzni az összeset).
- Figyelj rá, hogy a fájlnevek pontosan azok legyenek, amiket a feladat kér!
- A Moodle által használt fordító nem feltétlenül támogatja az újabb nyelvi fejlesztéseket. Amit az órákon tanultunk és használtunk, az mind működik, de az újabb fejlesztések nem biztos.

Feladat – Kisz 1

- A feladat elkezdéséhez a mellékelt projekt tartalmaz kódokat. A megadott **Program.cs** fájl a tesztelésben segít. Módosítani csak az elején lévő #define-okon lehet, amik a tesztek aktiválják.
- Az alapkódban adott egy **Merkozes** osztály, ami egy mérkőzést reprezentál kettő csapat között. Az osztály a két csapat nevét tárolja (hazai és vendég), amiket a konstruktor beállít, és Property-ken keresztül lekérdezhetők. **Az osztályt módosítani nem lehet!**
- A feladatban egy **MerkozesSorozat** osztályt kell írni, amit a **MerkozesSorozat.cs** fájlban kell feltölteni.
- A feladatok között lehet összefüggés, egy adott feladat teszteléséhez (és így elfogadásához) szükség lehet több előzőre is.

A feladatok:

1. Készíts egy **MerkozesSorozat** osztályt az **Merkozes**-ből származtatva.
 - a. Az osztály az eredeti adatokon felül tároljon még egész számokat egy dinamikus tömbben. Ezek a számok a két csapat között játszott meccsek eredményeit fogják jelölni (1, ha a hazai csapat nyert, 2, ha a vendég csapat, 0, ha döntetlen).
 - b. Az osztály konstruktora kapja meg a két csapat nevét, valamint a lejátszott meccsek számát.
 - c. A konstruktor gondoskodjon a megfelelő méretű tömb létrehozásáról. Létrehozáskor a tömb minden eleme 0 legyen.

(2 pont)
2. A **MerkozesSorozat** osztályban legyen egy **UjEredmeny** metódus, amely paraméterben egy indexet és egész számot vár (ilyen sorrendben). A metódus állítsa be az index-edik helyen lévő eredményt arra, amit a paraméterben kap.

(1 pont)
3. A **MerkozesSorozat** osztályban legyen egy **NyertesCsapat** metódus, amely visszaadja a sorozatot nyerő csapat nevét (aki többször nyert a kettő közül). Ha esetleg egyenlő lenne az állás, akkor a „Dontetlen” szöveget adja vissza.

(2 pont)
4. A **MerkozesSorozat** osztályban legyen egy **Kiir** metódus, amely megjeleníti az adatokat a minta szerint. Az első sorban a két csapat neve legyen, majd minden sorban egy meccs eredménye (a sor elején a sorszám 1-től kezdve, majd az eredmény).

(2 pont)
5. A **Kiir** metódusnak legyen egy opcionális logikai paramétere, amely alapértelmezetten hamis. Ha a paraméter hamis, akkor a kiírás az eddigieknek megfelelően történjen. Ha a paraméter igaz, akkor az eredmények megjelenítésekor a szám helyett a megfelelő csapat neve szerepeljen (1 esetén a hazai csapat neve, 2 esetén a vendég csapat neve, 0 esetén az, hogy „Dontetlen”).

(1 pont)
6. A **MerkozesSorozat** osztályban az **UjEredmeny** metódus csak akkor állítsa be az értéket, ha az index helyes. Legyen az osztályban egy statikus adattag, ami folyamatosan számolja, hogy a függvényt mennyiszor hívtuk meg érvénytelen indexszel (az összes objektumban összesen). Ez az érték legyen lekérdezhető a szintén statikus **RosszIndexek** Property segítségével.

(2 pont)