

Formula–1 bajnokság (40 pont)

Egy Formula–1-es bajnokságban versenyzők, csapatok és versenyek szerepelnek. Készítsd el a F1Exception, Versenyzo, Csapat és Bajnoksag osztályokat az alábbi leírás alapján!

● A F1Exception osztály (2 pont)

Hozz létre egy F1Exception nevű kivételosztályt, amely az Exception beépített kivételosztályból öröklődik!

Készíts egy egyparaméteres konstruktort az osztályon belül, amely egy hibaüzenetet vár paraméterben!

A konstruktorban hívd meg az őosztály konstruktorát, és add át neki paraméterben a hibaüzenetet!

👤 A Versenyzo osztály (13 pont)

1. Hozz létre egy Versenyzo osztályt, amely a nev, orszag és pontszam adattaggal rendelkezik!
 2. Az osztály konstruktora a versenyző nevét, országát és pontszámát várja paraméterben (ebben a sorrendben)!
 3. A pontszám paraméter alapértelmezett értéke legyen 0!
 4. Inicializáld az adattagokat a paraméterek alapján! (3 pont)
 5. Írj gettert és settert a nev adattaghoz! (1 pont)
 6. Írj gettert és settert az orszag adattaghoz! (1 pont)
 7. Írj gettert és settert a pontszam adattaghoz!
 - A setter csak 0 és 500 közötti egész számot engedélyezzen!
 - Hibás érték esetén dobj F1Exception kivételt a következő üzenettel: „**Érvénytelen pontszám!**” (2 pont)
 8. Írd felül a ToString() metódust úgy, hogy a következő formátumban térjen vissza:
{nev} ({orszag}) – {pontszam} pont (2 pont)
 9. Definiáld felül az == operátort úgy, hogy két versenyző akkor egyenlő, ha a nevük és országuk megegyezik (kis- és nagybetűk ne számítsanak)! (3 pont)
-

🏎️ A Csapat osztály (10 pont)

1. Hozz létre egy Csapat osztályt, amelynek neve és versenyzok nevű adattagja van (a versenyzők listában vannak)! (2 pont)

2. A konstruktor paraméterként kapja meg a csapat nevét, és inicializálja a versenyzok listát üresen! (1 pont)
 3. Készíts egy hozzáad metódust, amely egy Versenyzo objektumot ad a csapathoz!
 - Ha nem Versenyzo típusú objektumot kap, dobj TypeError kivételt a következő üzenettel: **„Nem versenyző!”** (2 pont)
 4. Készíts egy pontszam_osszesen metódust, amely visszaadja a csapat összpontszámát! (2 pont)
 5. Írd felül a ToString() metódust úgy, hogy a következő formátumban térjen vissza: **{csapatnév}: {versenyző1}, {versenyző2}, ...** (3 pont)
-

A Bajnoksag osztály (15 pont)

1. Hozz létre egy Bajnoksag osztályt, amelynek egy csapatok nevű listája van! (1 pont)
 2. A konstruktor inicializálja ezt a listát üresen! (1 pont)
 3. Készíts egy hozzáad_csapat metódust, amely egy Csapat objektumot ad a listához!
 - Hibás típus esetén dobj TypeError kivételt a következő üzenettel: **„Nem csapat!”** (2 pont)
 4. Készíts egy legjobb_csapat metódust, amely visszaadja azt a csapatot, amelynek a legnagyobb az összpontszáma! (3 pont)
 5. Készíts egy statisztika metódust, amely visszaadja, hogy hány versenyző van országonként!
 - Az országneveket kisbetűsre alakítva számold!
 - Az eredményt egy Dictionary<string, int> formában add vissza! (5 pont)
 6. Írd felül a ToString() metódust úgy, hogy az összes csapatot és versenyzőiket sorolja fel, új sorba tördelve! (3 pont)
-

Felhasználói felület (10 pont)

Készíts egy **Windows Forms** alkalmazást, amely lehetővé teszi:

- Új versenyzők hozzáadását egy csapathoz
- Csapatok létrehozását és bajnokságba illesztését
- A bajnokság állásának megtekintését (ListBox)
- A legjobb csapat és országonkénti statisztika megjelenítését (Label)