

Forma-1 történelmi adatbázis

Specifikáció

Egy Forma-1 történelmi adatbázis alkalmazást kell elkészíteni: versenyek eredményeit lehet bevinni, majd ezek segítségével világbajnoki szezonokról adatokat kinyerni.

Az alkalmazás egy fájlt fog beolvasni. Kimenete egyszerű szöveges, bemenete szigorúan definiált parancsformátum. Egy parancs részei pontosvesszővel kerülnek elválasztásra (akárcsak a CSV formátumnál), és a legelső része - a parancsszó - határozza meg a további részek - argumentumok - jelentését.

Lehetséges parancsok

RACE: egy verseny bevitele következik.

Argumentumai: évszám, nagydíj neve (pl. Italian Grand Prix), hányadik verseny az adott évben, pont szorzó.

A pont szorzó értéke lehet: 0, 1, 0.5 és 2. A megszerezhető pontszámot meg kell szorozni ezzel az értékkel a világbajnokság eredményének számításakor.

Ezután a parancs után kizárólag RESULT, FASTEST és FINISH parancsokat lehet kiadni.

RESULT: a legutóbbi RACE parancs után kiadva egy versenyző eredményeit lehet bevinni a RACE-hez tartozó versenyhez.

Argumentumai: célba érési pozíció, versenyző neve, csapat neve.

Csak RACE és FINISH parancs között lehet kiadni.

FASTEST: a legutóbbi RACE parancs után kiadva a leggyorsabb kör tulajdonosát lehet bevinni a RACE-hez tartozó versenyhez.

Argumentumai: versenyző neve, csapat neve.

Csak RACE és FINISH parancs között lehet kiadni.

FINISH: a verseny bevitele véget ért.

Nincsenek argumentumai.

Csak RACE parancs után lehet kiadni, ha azóta nem volt FINISH parancs, és legalább 10 RESULT parancs került kiadásra.

QUERY: lekérdezés fog következni egy adott évre vonatkozóan. Kétféle módon működik: az elsőben az adott év végeredményét, a második esetben az adott verseny befejeztét követő állást jeleníti meg.

Argumentumai 1: évszám.

Argumentumai 2: évszám, hányadik verseny.

Ezután a parancs után kizárólag POINT parancsot lehet kiadni.

POINT: a legutóbbi QUERY parancs után kiadva beállításra kerül, hogy a lekérdezésnél milyen pontszámítás történjen, ezután a lekérdezés végrehajtásra kerül.

Argumentumai: pontszámítási módszer.

Pontszámítási módszer lehetséges értékei: CLASSIC, MODERN, NEW, PRESENT

CLASSIC pontozás: 10, 6, 4, 3, 2, 1 pontot kap az első 6 helyezett ilyen sorrendben.

MODERN pontozás: 10, 8, 6, 5, 4, 3, 2, 1 pontot kap az első 8 helyezett ilyen sorrendben.

NEW pontozás: 25, 18, 15, 12, 10, 8, 6, 4, 2, 1 pontot kap az első 10 helyezett ilyen sorrendben.

PRESENT pontozás: 25, 18, 15, 12, 10, 8, 6, 4, 2, 1 pontot kap az első 10 helyezett ilyen sorrendben, a leggyorsabb kört futó pilóta szintén 1 plusz pontot kap, amennyiben az első 10 között végzett.

Csak QUERY parancs után lehet kiadni, ha azóta nem volt hozzátartozó POINT parancs.

Példa input

Egy példa input mellékletként átadásra kerül, mellyel tesztelni lehet az alkalmazást (egyéb teszteket is fogunk futtatni). A teszt valós adatokból készült, azaz a megfelelő szezon végeredményének (vagy köztes állásának) utánajárva ellenőrizni lehet a működést.

Output formátuma

Szabadon megválasztható, de a konzolra kell kiírnia, és világosnak és egyértelműnek kell lennie.

Amennyiben helytelen input / parancs érkezik, azt egyértelmű és világos hibaüzenettel kell jelezni a felhasználó felé.

Kikötések

Nem lehet felhasználni semmilyen Java keretrendszert vagy külső library-t. Így elég, ha az alkalmazás bármilyen megfelelő fordításon átesve parancssorból vagy IDE-ből futtatható (azaz nem szükséges Maven vagy Gradle build használata, JAR csomagolás, stb).

Java 8-ra készüljön az alkalmazás (AdoptOpenJDK-val lesz ellenőrizve).

Standard outputra kell írni, fájlból nem olvashat (kivéve az input fájlt) és fájlba nem írhat adatot az alkalmazás, emellett semmilyen más külső kapcsolatot nem létesíthet.

Tanácsok

Alaposan tervezd meg az alkalmazást előre, az objektumorientált elveknek megfelelően – ez az elsődleges szempont, amit nézni fogunk. Természetesen emellett hatékornak is kell lennie az alkalmazásnak.

Amennyiben a specifikációból hiányzik valami / nem egyértelmű valami, kérdezz utána: csendes.david@alerant.hu címen tudsz elérni.