A készítendő repülőgép forgalmi rendszerének egy modulja, amely a repülőjáratok jelenlegi állapotáról nyújt információkat

Az alábbi adatokat szeretnénk kezelni:

- Járatszám
- Kezdeti- és cél repülőtér
- Az indulás kiírt időpontja
- Aktuális késés

1.Feladat Hozz létre egy új projektet **JaratKezeloProject** néven. C# / Java / JavaScript / TypeScript nyelvet használj. A projekthez inicializálj egy Git repot is. A későbbi feladatok megoldását ebbe a repoba commitold. Az elkészült projektet töltsd fel GitHubra.

C# nyelv esetén a projekt típusa ClassLibrary legyen és .NET 8 vagy újabb keretrendszer verzióval dolgozz.

Java nyelv esetén használjon valamilyen függőség kezelőt (Maven vagy Gradle) a projekt létrehozásakor, valamint Java 17 vagy újabb JDK verzióval dolgozz.

JavaScript / TypeScript nyelv esetén a projektet inicializálja valamely csomagkezelő alkalmazás számára (npm / yarn)

a.) C# nyelv esetén adj hozzá a solutionhöz egy új nUnit vagy xUnit projektet **TestJaratKezeloProject** néven.

b.) Java vagy JavaScript / TypeScript nyelv esetén telepítsen automatizált teszteléshez szükséges könyvtárakat (javasolt packagek: java – junit | JavaScript – vitest)

2.Feladat A **JaratKezeloProject**-ben hozz létre egy osztályt **JaratKezelo** néven C# nyelv esetében az automatikusan generált osztályt nevezze át erre. Az osztályban az alábbi függvényeket valósítsd meg:

A függvények hibás paraméter esetén (nem létező járat, nem létező reptér, duplikált járat stb.) dobjanak:

C# nyelv esetén **ArgumentException**-t, Java nyelv esetén **IllegalArgumentException**-t, JavaScript nyelv esetén **Error**-t.

C# nyelv esetén a függvények nevét nagy betűvel szokás kezdeni.

a.) void ujJarat(string jaratSzam, string repterHonnan, string repterHova, DateTime indulas)

- Létrehoz egy új járatot, amit eltárol a járatkezelőben.
- Kezdetben nincs késés.
- A járatszámnak egyedinek kell lennie!
- DateTime típus helyett Java nyelvben LocalDateTime, JavaScript nyelvben Date típust használjon.

b.) void keses(string jaratSzam, int keses)

- A megadott késést hozzáadja a megadott járathoz.
- A keses paraméter percben értendő.
- Ha többször hívjuk meg a függvényt a késések összeadódnak
- A keses paraméter negatív is lehet, ez azt jelenti, hogy egy korábban rögzített késést sikerült elhárítani, behozni.
- A szumma késés, azaz a Keses() függvényből összeadódó érték viszont sosem lehet negatív, a gép nem indulhat korábban, mint a kiírt időpont. Ha mégis az lenne, akkor dobjunk **NegativKesesException** -t! (JavaScript esetén a nyelv működése miatt elég sima Error-t dobni)
- *Szorgalmi – nehezítés: int paraméter helyett a késést C# nyelvben TimeSpan, Java nyelvben pedig Duration objektummal add meg*

c.) DateTime mikorIndul(string jaratSzam)
- Meghatározza, hogy az adott járatszámú gép mikor indul ténylegesen.
- Az indulás időpontjához hozzáadja a késést.
- DateTime típus helyett Java nyelvben LocalDateTime, JavaScript nyelvben Date típust használjon.

d.) List<string> jaratokRepuloterrol(string repter)
- Azon járatok járatszámát adja vissza, amelyek a paraméterben megadott nevű repülőtérről indulnak.

3.Feladat Teszteld a JaratKezelo osztályt automatizált tesztekkel. Legyen benne a helyes kimenet ellenőrzése, és a helytelen bemenetekre kapott kivételek ellenőrzése is. A teszteket egy külön fájlban (és amennyiben a teszt keretrendszer megköveteli külön osztályban) készítsd el, melynek nevével utalj rá, hogy az JaratKezelo osztály teszteseteit tartalmazza. A fájl (és osztály) elnevezésekor ügyelj a választott nyelv elnevezési konvencióira.

JaratKezelo osztály uml diagrammja: