# **Program Specifikáció**

## 1. Célkitűzés

Ez a program célja egy adatfeldolgozó rendszer kialakítása, amely több különböző típusú adatot *(Auto, Felszereltseg, Kategoria, Felhasznalo, Szemely, Lezart\_berles)* CSV fájlokból **olvas** be, majd az adatokat strukturáltan **tárolja**, hogy azokat később könnyen el lehessen érni és **feldolgozni**.

A programnak hatékonyan kell kezelnie az adatbeolvasást és feldolgozást, mivel összesen 8300 adatsor feldolgozása szükséges. Ennek érdekében olyan optimalizációs technikák kerülnek alkalmazásra, melyek a beolvasás idejére és az erőforrásigényre optimalizáltak.

## 2. Adatsorok és Fájlok

A program a következő **CSV** fájlokból olvassa be az adatokat:

* **Autok.csv** – 550 adatsor
* **Felszereltségek.csv** – 550 adatsor
* **Kategoriak.csv** – 550 adatsor
* **Felhasznalok.csv** – 1450 adatsor
* **Szemelyek.csv** – 1450 adatsor
* **Lezart\_berlesek.csv** – 3750 adatsor

**Összes adatsor**: 8300 db

## 3. Adatszerkezetek és Működés

Minden fájlstruktúra egy saját ***record*** osztállyal rendelkezik, amely egy statikus Beolvasas metódust is tartalmaz. Ez a **metódus** a sorról-sorra való **beolvasást** végzi, majd a CSV sorból **létrehozza** az adott rekord típusú **objektumot**. Például:

* ***Auto.cs*** *– Az Auto osztályban a Beolvasas metódus egy-egy autót reprezentáló adatokat dolgoz fel*.
* ***Felszereltseg.cs****,* ***Kategoria.cs****,* ***Felhasznalo.cs****,* ***Szemely.cs****,* ***Lezart\_berles.cs*** *–* Ezek **mind** hasonló módon rendelkeznek **saját** **Beolvasas** **metódussal**, amely az adott fájltípushoz kapcsolódó adatokat dolgozza fel.

A ***DataStore*** osztály **kezeli** a **beolvasást** és az adatokat egy-egy megfelelő listába helyezi, így biztosítva az adatokhoz való gyors hozzáférést.

## 4. Beolvasást Gyorsító Optimalizációk

Az alábbi optimalizálási technikák kerültek bevezetésre a beolvasási folyamat gyorsítása érdekében:

### StreamReader használata File.ReadLines helyett

* A StreamReader közvetlenül olvassa be a fájlokat, és nagyobb puffereléssel hatékonyabbá teszi a beolvasást. A File.ReadLines helyett StreamReader használatával a beolvasási művelet gyorsabb és kevésbé terheli a rendszert.

### Párhuzamos Feldolgozás (Parallel feldolgozás)

* A DataStore osztály a Parallel.Invoke metódust használja, hogy minden fájl beolvasása egyszerre, párhuzamosan történjen. Ez a technika kihasználja a többmagos processzorok nyújtotta teljesítményelőnyöket, így jelentősen csökkenti a beolvasás idejét.

### Fájlok külön metódusban történő feldolgozása

* Minden egyes fájlhoz külön Beolvas... metódus készült, amelyek az adott típusú fájl beolvasásáért felelnek (például BeolvasAutok, BeolvasFelszereltsegek). Ezáltal minden fájlbeolvasás célzott és egyértelmű, elkerülve az általános, komplex beolvasási logikát.

## 5. Program Használata és Funkciói

### **DataStore beolvasás**:

A DataStore osztály konstruktorában automatikusan meghívja az összes beolvasó metódust, így az összes fájlból az adatok betöltődnek a megfelelő listákba.

### **Lekérdezések a CLI felületen:**

A CLI felületen keresztül különböző lekérdezéseket futtathatunk az adatokra, például:

Adatsorok száma: A DataStore példányban lévő adatok száma egyszerűen elérhető és megjeleníthető (pl. Autok.Count()).

Gyártók és típusok listázása: Az autók listázása gyártó és teljesítmény szerint csoportosítva.

VW autók száma: A CLI felület lekérdezéseivel az összes VW márkájú autó száma is megjeleníthető.

### Felhasználói szűrés:

A program felhasználói input alapján képes a különböző paraméterekre szűrni (pl. Típus, Márka, Teljesítmény). Az ékezetek és kisbetűk figyelmen kívül hagyásával a program rugalmas a különféle beviteli formátumokra.

### **Felhasználói Input és Szűrési Lehetőségek:**

A felhasználó különböző szempontok alapján szűrheti az autóflottát, mint például Típus, Márka, Teljesítmény, Hatótáv, Rendszám, Gyártási év, Futásteljesítmény.

A program támogatja a kis- és nagybetűkre, valamint ékezetekre érzéketlen szűrést, hogy a bevitel rugalmas legyen. Emellett rövidítéseket is elfogad a szűrési szempontokhoz.

### Szűrési szempontok kezelése:

A szűrési szempontok egyszerűsítve is beírhatók (pl. "típus", "teljesítmény", "rendszam", "futás"), és a program ezeket különböző változatokban elfogadja (ToLower és RemoveAccents segítségével), hogy a felhasználó számára kényelmesebbé tegye az interakciót.