**Compilers Documentation**

Az első feladatom a félkész console-os alkalmazásunk in-outputjának vizuális megjelenítése, ehhez hozzáadtam egy **WinForms** -t a már meglévő projecthez.

Ezt nem a project forráskódjának átmásolásával egy új projectbe értem el, hanem indítottam egy külön szálat a winforms-nak, így a program indításakor mind a console-os felület, mind a winForm-s elérhető számunkra.

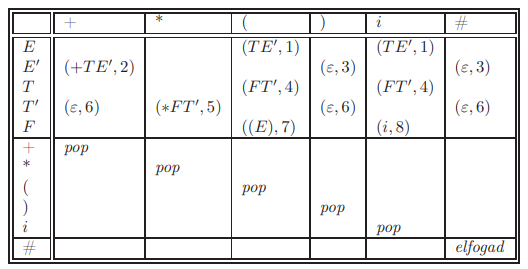
**Célok/Terv :**

**Funkciók:** Adatbázis beolvasása, az alkalmazásban készített adatbázis kimentése () fájlformátumba.

Tetszőleges méretű szabály táblázat generálása, kitöltése, majd a program ezt értékelje ki.

Regexxel az [0-9], [a-z], [A-Z] inputok a nyelvtanunk által meghátorozott tetszőleges karakterré alakuljanak (ez a mi példánkban az **i** volt).

A lefutás lépésről való leírása és/vagy egyből kiirja hogy sikeres e ez vagy válaszható avgy mind2t csinálja.

Az output célját megerősítő színnel megjleneíteni (false: piros, sikeres volt: zöld)

Responsive, egyértelmű, jól átlátható design/ megjelenés.

hashmarkkal kiegészíti a lekérdezést.

hülye biztos, le vna kezelve ez az

először a jsonnel, majd a xml el próbálkoztam, mind tul bonyolultan, serailizekkel mukodott, a xml ben pedig tag nevekként semiképpen sem megadható speciális karakter, ami a mi esetünkben pedig szükséges lenne (+,\*) input jellel végezhető műveletek.

Ezért a választásom a csv file kiterjesztésre esett.

PM> Install-Package CsvHelper

de ez nem is kell

Szoveget beolvasom plain etxtkent, aztan datatableve alakitom es azt adom at a datagridviewnek datasourcekent.

Lehetőségünk van az output egybőli szerkeztésére is, kényelmesen, egy szerkezthető textboxban mindent megtalálunk.

Messageboxokkal usernrek visszajelzesek

„Okos” file mentés , ha egyből mentünk, temp.csv b menti ha nem akkor a kiböngészett helyen írogatja felül a fileunkat.

Amíg a felhasználó nem töltötte be a szabály táblázatot, addig nem tud analizálni sem.