



EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

ALGORITMUSOK ÉS ALKALMAZÁSAIK TANSZÉK

## Intelligens szövegszerkesztő

*Témavezető:*

Dr. Szabó László Ferenc

tanszékvezető egyetemi docens

*Szerző:*

Bene Fruzsina

programtervező informatikus BSc

*Budapest, 2022*

# Tartalomjegyzék

|   |           |
|---|-----------|
| <b>1. Bevezetés</b>                                   | <b>4</b>  |
| <b>2. Elméleti áttekintés</b>                         | <b>6</b>  |
| 2.1. .NET keretrendszer . . . . .                     | 6         |
| 2.2. Windows Presentation Foundation . . . . .        | 6         |
| 2.3. eXtensible Application Markup Language . . . . . | 7         |
| 2.4. WPF RichTextBox vezérlő . . . . .                | 7         |
| 2.4.1. FlowDocument . . . . .                         | 8         |
| 2.5. Levenshtein távolság . . . . .                   | 8         |
| <b>3. Felhasználói dokumentáció</b>                   | <b>12</b> |
| 3.1. A program ismertetése . . . . .                  | 12        |
| 3.2. Rendszerkövetelmények . . . . .                  | 13        |
| 3.3. Telepítés és indítás . . . . .                   | 13        |
| 3.4. Kezdőoldal . . . . .                             | 14        |
| 3.5. Főoldal . . . . .                                | 15        |
| 3.5.1. Oldalbeállítások menüpont . . . . .            | 19        |
| 3.5.2. Formázás menüpont . . . . .                    | 21        |
| 3.5.3. Beszúrás menüpont . . . . .                    | 23        |
| 3.5.4. Szótár menüpont . . . . .                      | 24        |
| 3.6. Szótár . . . . .                                 | 25        |
| 3.7. "Erre gondolt?" . . . . .                        | 25        |
| 3.7.1. Szótár létrehozása . . . . .                   | 26        |
| 3.7.2. Szótár betöltése . . . . .                     | 26        |
| 3.8. Mentés, betöltés . . . . .                       | 26        |
| 3.9. Egyéb információk . . . . .                      | 27        |
| 3.10. Gombok jegyzéke . . . . .                       | 27        |

|   |           |
|---|-----------|
| <b>4. Fejlesztői dokumentáció</b>                     | <b>29</b> |
| 4.1. Specifikáció . . . . .                           | 29        |
| 4.1.1. Feladat . . . . .                              | 29        |
| 4.2. A rendszer architektúrája . . . . .              | 29        |
| 4.2.1. Fejlesztői környezet . . . . .                 | 30        |
| 4.2.2. A forráskód beszerzése . . . . .               | 30        |
| 4.2.3. Könyvtárszerkezet . . . . .                    | 30        |
| 4.2.4. Névválasztási és kódolási konvenciók . . . . . | 31        |
| 4.2.5. Az alkalmazás struktúrája . . . . .            | 31        |
| 4.3. Felhasználói felület . . . . .                   | 32        |
| 4.3.1. Kezdőoldal . . . . .                           | 33        |
| 4.3.2. Az alkalmazás főoldala . . . . .               | 33        |
| 4.3.3. Dialógus ablakok . . . . .                     | 33        |
| 4.4. Felhasználói súgó . . . . .                      | 34        |
| 4.5. Osztályszerkezet . . . . .                       | 34        |
| 4.5.1. A nézet felépítése . . . . .                   | 35        |
| 4.5.2. Perzisztencia . . . . .                        | 36        |
| 4.5.3. A .doced fájl tartalma . . . . .               | 37        |
| 4.5.4. A modell felépítése . . . . .                  | 37        |
| 4.5.5. A nézetmodell felépítése . . . . .             | 41        |
| 4.5.6. Az elemző felépítése . . . . .                 | 47        |
| 4.5.7. Környezet . . . . .                            | 52        |
| 4.6. Felhasználói esetek . . . . .                    | 53        |
| 4.7. Felhasználói történetek . . . . .                | 54        |
| 4.8. Tesztelés . . . . .                              | 67        |
| 4.8.1. Tesztkörnyezet . . . . .                       | 67        |
| 4.8.2. Manuális tesztelés . . . . .                   | 68        |
| 4.8.3. Unit tesztek . . . . .                         | 69        |
| 4.9. Verziókezelés . . . . .                          | 71        |
| <b>5. Összegzés</b>                                   | <b>72</b> |
| 5.1. Továbbfejlesztési lehetőségek . . . . .          | 72        |
| <b>A. Súgó a szótár használatához</b>                 | <b>74</b> |

|                   |    |
|-------------------|----|
| Irodalomjegyzék   | 76 |
| Ábrajegyzék       | 78 |
| Táblázatjegyzék   | 80 |
| Algoritmusjegyzék | 81 |
| Forráskódjegyzék  | 82 |

# 1. fejezet

## Bevezetés

Digitálisan dokumentumokat készíteni kényelmes, hasznos és sok esetben elkerülhetetlen is tanulmányaink folyamán. Legyen szó órai jegyzetokről, dolgozatokról, beadandókról, házi feladatokról, digitálisan könnyen átlátható és később is jól szerkeszthető dokumentumokat készíthetünk. Amikor létre szeretnénk hozni egy dokumentumot, számtalan különböző szövegszerkesztő alkalmazás közül válogathatunk attól függően, hogy mi a dokumentumunk témája, típusa, mik a saját igényeink. Azonban amikor egy matematikai jegyzetet vagy más matematikai témájú dokumentumot szeretnénk elkészíteni több problémába is ütközhetünk a megfelelő alkalmazás vagy környezet kiválasztása közben. A ma használt népszerű szövegszerkesztő alkalmazások, mint a Microsoft Word, Apple Pages, LibreOffice Writer megfelelő funkciókat nyújtanak, azonban használatuk sokszor nem kényelmes.

A kényelmetlenség leginkább abban rejlik, hogy a sokszor előforduló azonos formázást igénylő részeket minden egyes előfordulásukkor újra manuálisan formázni kell, vagy egy előző előfordulásukat másolni, és beilleszteni. Ez jelentősen lassítja a dokumentum szerkesztését. Másik lehetőség LaTeX dokumentumok készítése, azonban ehhez jelentős mértékű előzetes tudásra van szükség, és a LaTeX használata sem tud közel sem olyan gyors lenni még azoknak sem, akik nagyon jól ismerik, mintha csak a dokumentum szövegét gépnénk le. A gyorsaság viszont sok esetben nagyon jól tud jönni.

Jelen dolgozat keretein belül erre a problémára szeretnék egy megoldást adni, egy könnyen kezelhető és kényelmes szövegszerkesztő alkalmazás létrehozásával. Az alkalmazás tökéletesen használható matematikai vagy akár más tartalmú jegyzetek létrehozására. Az alkalmazás alapötlete az, hogy az ismétlődő formázási mintákat

elmenti, és az azonos szöveg következő előfordulásakor automatikusan ráteszi a korábbi formázást, amit legnagyobb valószínűséggel szeretne a felhasználó is alkalmazni rá. Ezáltal a dokumentum szerkesztése gyors és kényelmes lesz.

## 2. fejezet

# Elméleti áttekintés

### 2.1. .NET keretrendszer

A .NET platform a Microsoft szoftverfejlesztési platformja, ami több keretrendszert és programcsomagot tartalmaz. A .NET Standard alapra épül egységesen, és többplatformos szoftverfejlesztést tesz lehetővé. [1]

1998-tól a futásidejű fordítás kiterjesztése volt a Microsoft célja a létező Microsoft nyelvekre, ami akkoriban a Visual C++ és Visual Basic voltak. Ezáltal azt akarták lehetővé tenni, hogy egy közös virtuális gépen futhassanak a programok, bevezethessenek egy egységes köztes nyelvet, amivel a programozási nyelvek szabadon kombinálhatók egymással, és használhatók legyenek egységes programkönyvtárak. [2]

Így készült el a .NET Framework első béta verziója 2001-re és a dedikáltan ráépülő nyelv, a C#. Ez azóta integrált része a Windows rendszereknek. 2016-ban megjelent a .Net Core, ami a .NET Framework problémáira nyújt megoldást. Ez egy cross-platform, nyílt forráskódú keretrendszer, ami a korábbi zárt forráskódú, Windows-központú, monolitikus felépítésű keretrendszerhez képest jelentősen több lehetőséget nyújt.

### 2.2. Windows Presentation Foundation

Az alkalmazásomat .NET 5 keretrendszerben fejlesztettem WPF alkalmazásként. A Windows Presentation Foundation (WPF) a .NET környezet csak Windowson elérhető vektoros alapú grafikus felületi rendszere, ami lehetővé teszi a grafikus kártyák kihasználását, lehetőséget ad a felület deklaratív leírására XAML nyelv segítségével,

és az MVVM architektúra támogatásával függetleníti a megjelenést és a vezérlést, így jelentősen javít az alkalmazás architektúráján.

## 2.3. eXtensible Application Markup Language

Az eXtensible Application Markup Language (XAML) olyan XML alapú deklaratív nyelv, mely biztosítja a grafikus felület teljes leírását, lehetőséget adva két- és háromdimenziós elemek, animációk és effektek leírására.

## 2.4. WPF RichTextBox vezérlő

A dolgozatomhoz szükségem volt egy vezérlőre, amin a szövegek szerkesztését meg tudom valósítani, el tudom menteni és be tudom tölteni a szerkesztett szöveget. Hosszas kutatás után a RichTextBox<sup>1</sup> vezérlőre esett a választásom, ami a legtöbb szempontból megfelelő választásnak bizonyult.

A dokumentum maga egy FlowDocument<sup>2</sup> osztálypéldány, amihez a RichTextBox tökéletes vezérlő felületet biztosít, ugyanis nem csak a FlowDocument olvasását, hanem annak szerkesztését is lehetővé teszi. Ez az egyetlen olyan vezérlő FlowDocument-hez, amivel annak szerkesztése is lehetővé válik. A RichTextBox többek között olyan funkciókat támogat, mint a helyesírás-ellenőrzés, billentyűkombinációk használata szöveg formázásához, képek, bekezdések, táblázatok, felsorolások megjelenítése.

A FlowDocument lehetőséget ad többoldalas és több egymás mellett elhelyezett hasábot tartalmazó dokumentumok szerkesztésére is, azonban a RichTextBox ezeket nem támogatja. Ennek ellenére a RichTextBoxban szerkesztett FlowDocument is megjeleníthető más vezérlőkkel többoldalas nézetben, így tetszőlegesen felépített dokumentum teljeskörű szerkesztésére alkalmas a vezérlő.

A RichTextBox-ban szereplő teljes szöveg bejárható TextPointerek segítségével. Az adott pozícióhoz tartozó TextPointert mindig a FlowDocument kezdőpozíciójá-

---

<sup>1</sup><https://docs.microsoft.com/en-us/dotnet/desktop/wpf/controls/richtextbox-overview?view=netframeworkdesktop-4.8>

<sup>2</sup><https://docs.microsoft.com/en-us/dotnet/desktop/wpf/advanced/flow-document-overview?view=netframeworkdesktop-4.8>



tól kell kiszámítani. A kurzor vagy a kijelölt részek pozíciói könnyen lekérdezhetők. A dokumentum szövegének egyes részein TextRange-eknek tekintve karakterről karakterre végig lehet iterálni, közben minden információt lekérdezhetünk vagy beállíthatunk az egyes karakterekre kód szinten. Ez a programomban nagy szerepet kapott, az elemzőhöz ilyen módon olvasom be a kódban a szavakat és a rájuk vonatkozó formázási információkat, és a formázások alkalmazásakor is ezt a módszert használom.

### 2.4.1. FlowDocument

A FlowDocument úgynevezett Block-okból áll, ez a közös elnevezése minden elemnek, ami a FlowDocumenten belül található. A dokumentum Paragraph-okra bontható, azokon belül pedig Run-ok találhatók. Ezek jól körülhatárolható elemek, amiknek a tulajdonságainak a lekérdezése és a manipulálása mind a felhasználó részéről az alkalmazás használata közben, mint kód szinten megengedett.

## 2.5. Levenshtein távolság

Az elemző fontos része az "Erre gondolt?" funkció, ami a szótárban nem szereplő szavakra ad meg lehetséges alternatívákat a szótárból. Mivel itt az alapfeladat két szöveg eltérésének a mértékének a meghatározása, erre kellett valamilyen hatékony megoldást találnom.

Egy, a gyakorlatban gyakran használt, jól bevált algoritmus a probléma megoldására a két szó Levenshtein távolságának [3] kiszámítása. A Levenshtein távolságot a megalkotója, egy orosz tudós, Vladimir Levenshtein után nevezték el, aki 1965-ben készítette el az algoritmust. Ez a távolság egy "edit distance", vagyis szerkesztési távolság. Minél nagyobb a távolság, a két szó annál inkább különbözik egymástól, és fordítva is igaz, minél kisebb a távolság, a két szó annál jobban hasonlít egymásra.

Az algoritmus alapja annak kiszámítása, hogy hány művelet elvégzésével tudjuk átalakítani az egyik szót a másikká. Minden művelet értéke 1-nek felel meg a távolság számításában.

A megengedett műveletek a karakterekre:

- törlés
- beszúrás
- csere.

Két szó között a Levenshtein távolság kiszámítása a következő rekurzív képlettel történik.<sup>3</sup>

$$\text{lev}_{a,b}(i, j) = \begin{cases} \max(i, j) & \text{if } \min(i, j) = 0, \\ \min \begin{cases} \text{lev}_{a,b}(i-1, j) + 1 \\ \text{lev}_{a,b}(i, j-1) + 1 \\ \text{lev}_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

2.1. ábra. Rekurzív formula a Levenshtein távolság kiszámításához

A távolság papíron történő kiszámítására az egyik legjobb módszer, amikor egy táblázat segítségével számítjuk ki az egyes részzavakra a távolságokat, részproblémákra bontva a feladatot, majd azokból lépésenként tovább haladva, az előző részprobléma megoldásánál kapott eredményeinket felhasználva végül megkapjuk a távolságot a teljes szavakra. Ez egy dinamikus programozás algoritmus.

A táblázatot a következőképpen rajzoljuk fel kezdetben:

|   |   | B | A | C | K | U | P |
|---|---|---|---|---|---|---|---|
|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| K | 1 |   |   |   |   |   |   |
| U | 2 |   |   |   |   |   |   |
| P | 3 |   |   |   |   |   |   |
| A | 4 |   |   |   |   |   |   |
| C | 5 |   |   |   |   |   |   |

2.1. táblázat. Levenshtein távolság táblázattal - első lépés

Az első sorban és oszlopban az egyes részzavak üres szóval vett távolságát számítjuk ki. Ezután a többi cellának az értékét úgy számítjuk ki, hogy a vele balra, fel, és átlósan fel balra levő cellák értékeinek vesszük a minimumát, és ha a két utolsó betű a részzavakból nem egyezik meg, akkor még hozzáadunk 1-gyet.

Ez alapján a további kitöltése a táblázatnak:

---

<sup>3</sup><https://www.baeldung.com/cs/levenshtein-distance-computation>

|   |   | B | A | C | K | U | P |
|---|---|---|---|---|---|---|---|
|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| K | 1 | 1 | 2 | 3 | 3 | 4 | 5 |
| U | 2 | 2 | 2 | 3 | 4 | 3 | 4 |
| P | 3 | 3 | 3 | 3 | 4 | 4 | 3 |
| A | 4 | 4 | 3 | 4 | 4 | 5 | 4 |
| C | 5 | 5 | 4 | 3 | 4 | 5 | 5 |

2.2. táblázat. Levenshtein távolság táblázattal

Végül az eredményt a táblázat jobb alsó sarkából tudjuk kiolvasni. Tehát a "BACKUP" és "KUPAC" szavak Levenshtein távolsága 5.

A távolság kiszámítására több megközelítést használhatunk. Ezek eltérnek egymástól az implementációjuk nehézségében és a hatékonyságukban. A legegyszerűbb és legkevésbé hatékony megvalósítás a rekurzív algoritmus megadása a Levenshtein távolság kiszámítására. Ez az algoritmus a hasonló prefixekre többször kiszámolja a távolságot, emiatt rendkívül költséges lesz.

---

### 1. algoritmus Rekurzív algoritmus Levenshtein távolság kiszámítására

---

**Funct** RecLevDist( $a, b$ )

```

1: if  $a.length() == 0$  then
2:   return  $b.length()$ 
3: end if
4: if  $b.length() == 0$  then
5:   return  $b.length()$ 
6: end if
7: if  $a[0] \neq b[0]$  then
8:    $c \leftarrow 1$ 
9: else
10:   $c \leftarrow 0$ 
11: end if
12: return
13:   $\min(\text{RecLevDist}(a.substring(1), b) + 1,$ 
14:     $\text{RecLevDist}(b.substring(1), a) + 1,$ 
15:     $\text{RecLevDist}(a.substring(1), b.substring(1)) + c)$ 

```

---

A papíron történő kiszámításhoz legközelebb a következő, iteratív algoritmus áll, ami hatékonyságával felülmúlja a rekurzív változatot. Itt is egy mátrix sorait és oszlopait számítjuk ki a képlet alapján, majd abból olvassuk ki az eredményt. Ez az implementáció a Wagner-Fischer algoritmusként ismert. A programomban én is ezt az algoritmust használom.

---

**2. algoritmus** Iteratív algoritmus mátrixszal

---

**Funct** MatrLevDist( $a, b$ )

```
1:  $distMatrix[i, 0] \leftarrow i \forall i \in 0..a.length()$ 
2:  $distMatrix[0, j] \leftarrow j \forall j \in 0..b.length()$ 
3: for  $j \leftarrow 1, \dots, b.length()$  do
4:   for  $i \leftarrow 1, \dots, a.length()$  do
5:     if  $a[i] \neq b[j]$  then
6:        $c \leftarrow 1$ 
7:     else
8:        $c \leftarrow 0$ 
9:     end if
10:     $distMatrix[i, j] \leftarrow \min($ 
11:       $distMatrix[i - 1, j] + 1$ 
12:       $distMatrix[i, j - 1] + 1$ 
13:       $distMatrix[i - 1, j - 1] + c)$ 
14:   end for
15: end for
16: return  $distMatrix[a.length(), b.length()]$ 
```

---

A Levenshtein távolság algoritmusának sok gyakorlati alkalmazása ismert. Felhasználják többek között helyesírás javító algoritmusokban, beszédfelismerő algoritmusokban és DNS elemzésnél. Ezekben az alkalmazásokban az algoritmus különböző változatait használják. A klasszikus algoritmust úgy módosítják, hogy az egyes műveletek különböző költségekkel rendelkeznek, ezáltal pontosabb eredményeket kaphatunk különböző körülmények között.

A program a szótárban nem szereplő begépelt szavakra lefuttatja az algoritmust a Levenshtein távolság kiszámításához minden a szótárban már szereplő szóra. Ha a kapott eredmény elég kicsi (maximum 2), akkor azt a szótár elemet egy listába fűzi, és a 3 első találatot felajánlja a felhasználónak. Ezek közül a felhasználó aztán kiválaszthatja a neki legmegfelelőbbet, vagy elutasíthatja a javaslatokat. Ha kiválaszt egyet a listából, akkor az utoljára begépelt szó kijavítódik a kiválasztottra, és a hozzá tartozó formázás alkalmazódik rá.

## 3. fejezet

# Felhasználói dokumentáció

Az alábbi felhasználói dokumentáció egy útmutató a program beszerzéséhez, telepítéséhez, indításához és használatához.

### 3.1. A program ismertetése

A program egy szövegszerkesztő alkalmazás, ami megfelelő környezetet kínál dokumentumok, szövegek, jegyzetek gyors és kényelmes szerkesztéséhez. Bizonyos dokumentumok szerkesztése közben előfordul, hogy adott szavakra ugyanazt a formázási stílust szeretnénk alkalmazni a szó összes vagy szinte összes előfordulása-kor. Ilyenkor sok ismételt műveletet jelent a szóra beállítani mindig újra és újra a formázásokat, ami jelentősen megnöveli a szövegírás idejét. Ez például egy egyetemi órai jegyzet készítése közben különösen rosszul tud jönni, mivel ilyenkor fontos, hogy minél gyorsabban írjuk a jegyzetünket, ne maradjunk le.

Erre a problémára kínál megoldást a jelen program. A dokumentum szerkesztése közben fel tudunk venni az úgynevezett szótárba készre formázott szavakat, és ezután egy elemző figyel az, amit begépelünk. Ha egyezést talál a szótár szavaival, akkor automatikusan alkalmazza rá a korábban beállított formázási stílust. Így a dokumentumunk jól formázott lesz, szinte gépelési sebességgel, csak a szerkesztés kezdetekor kell a szótárba felvennünk elemeket. Emellett az elemző az egyes elgépeléseinket is ki tudja szűrni, ekkor a szótárból ajánl fel csere lehetőségeket.

## 3.2. Rendszerkövetelmények

A program platformspecifikus, csak Windows operációs rendszeren futtatható. Az alkalmazás futtatásához optimális a Windows 7, vagy annál újabb operációs rendszer. A futtatható állomány x32 és x64-es architektúrájú rendszerekkel is használható.

Az alkalmazás hardware követelményeit a Visual Studio Diagnostic Tool segítségével állapítottam meg. Az alkalmazás alacsony erőforrásigényű, átlagos felszereltségű számítógépeken is jól működik.

Minimális hardware követelmények:

- 1024×768-as felbontású kijelző
- 200 MB szabad RAM
- 50 MB szabad háttértár
- Windows 7 x32 operációs rendszer

Ajánlott hardware követelmények:

- 1920x1080-as felbontású kijelző
- 8 GB RAM
- 200 MB szabad háttértár
- Legalább Windows 8 x64 operációs rendszer

## 3.3. Telepítés és indítás

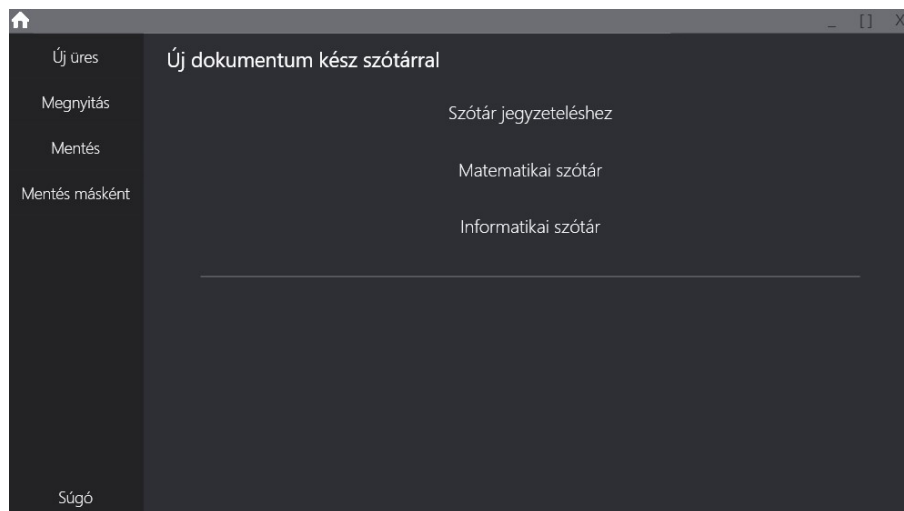
A program futtatásához szükséges fájlokat egy zip tömörített archívumként lehet letölteni. A fájlok kicsomagolása után a *DocEditor.exe* állományt kell elindítani.

A szoftver futtatásához szükséges, hogy a számítógép rendelkezzen telepített .NET 5 keretrendszerrel. Ennek a telepítése a következő linkkel érhető el.

- <https://download.visualstudio.microsoft.com/download/pr/14ccbee3-e812-4068-af47-1631444310d1/3b8da657b99d28f1ae754294c9a8f426/dotnet-sdk-5.0.408-win-x64.exe>

## 3.4. Kezdőoldal

A program indítása után először a kezdőképernyőt látjuk magunk előtt. Ez az oldal új dokumentum létrehozására, betöltésére szolgál. A felület tartalmaz egy oldalsó menüt és egy listát, ami betölthető fájlokat tartalmaz.



3.1. ábra. Az alkalmazás kezdőoldala

A kezdőoldalról elérhető a felhasználói súgó is, ami rögtön segítséget nyújt az első lépésekben. A súgó gombra kattintva megnyílik a felhasználói súgóhoz tartozó weboldal, ahol a tartalomjegyzék pontjain végignavigálva megismerhetjük a program használatát, működését.

Az "Új üres" gombra kattintva létrehozunk egy új üres dokumentumot, szótár nélkül. Így "tisztta lappal" kezdhetünk neki a dokumentumunk szerkesztésének, és a hozzá tartozó szótárat is teljes mértékben személyre szabhatjuk.

Az megnyitás gombot választva betölthetünk egyet a korábban szerkesztett fájlok közül, ennek segítségével könnyen tudjuk folytatni a legutóbb félbehagyott munkánkat.

A mentés és mentés másként gombok az alkalmazás indításkor nem használhatók, ezek csak a főoldalról megnyitott kezdőoldalról elérhetők.

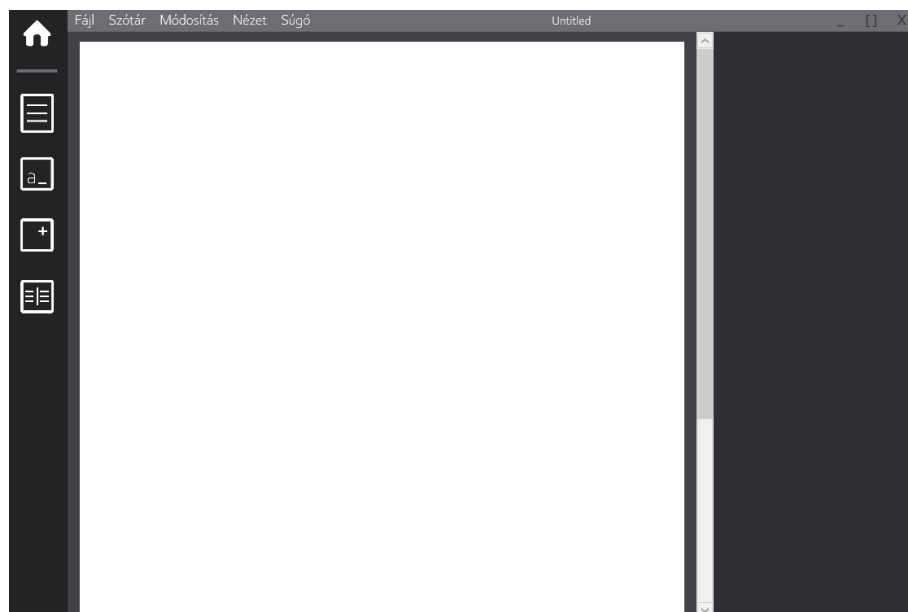
A kezdőoldalon levő listából egy korábban létrehozott szótárat választhatunk ki, amit egy új üres dokumentum mellé töltünk be.

A program kínál előre elkészített szótárakat is, amik adott témakörhöz hasznos formázott szavakkal vannak feltöltve. Ezekkel egy jó alappal tudjuk elkezdni a dokumentumunknak a szerkesztését. Három betölthető szótár közül választhatunk, ezek általános jegyzetek készítéséhez, matematikai és informatikai jegyzetekhez alkalmazhatók.

Miután kiválasztunk a listából egy betöltendő fájlt, vagy a bal oldalon található menüsor segítségével új üres fájlt hozunk létre, vagy megnyitunk egy korábbi fájlt, a kezdőoldaltól az alkalmazás főoldalára jutunk. A kezdőoldalt később is bármikor el tudjuk érni a főoldalról.

## 3.5. Főoldal

A főoldalon bal oldalon az alkalmazás menüjét láthatjuk, a szerkesztéssel kapcsolatos funkciókat nyithatjuk meg a segítségével. A főoldalon található gombok, vezérlők többségben ikonok alapján tájékoztatják a felhasználót az adott gomb funkcióját illetően. A főoldalon található vezérlők használata egyértelmű és könnyen értelmezhető bárki számára, emellett a gombokról és funkciójukról a későbbiekben részletes leírást is találunk.



3.2. ábra. Amit a főoldalon látunk a megnyitása után



A bal oldali menüben választhatjuk ki azt, hogy melyik szerkesztési menüpont nyíljon meg a jobb oldalon erre fenntartott területen. Ez a dokumentumszerkesztés menüje.

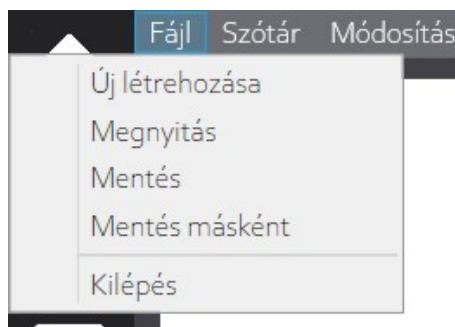


3.3. ábra. Az alkalmazás menüje

A menüpontok között rájuk kattintva lehet navigálni. A felső menüpont kiválasztásával az oldalbeállítások nyílnak meg, a második menüpont a formázási lehetőségeket nyitja meg, a harmadik felel a képek beillesztéséhez tartozó felület megjelenítéséért, az utolsó pedig a szótárért. Egyszerre egy menüpont választható ki.

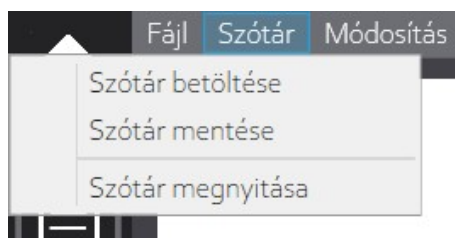
A felső házikót ábrázoló ikonra kattintással megnyithatjuk a kezdőoldalt. Ilyenkor csak a kezdőoldalon tudunk kattintani, azt bezárva visszajutunk a főoldalra, ami változatlan maradt a kezdőoldal megnyitása óta.

Az alkalmazás felső menüsorában található az alkalmazáshoz tartozó általános műveletek, mint a mentés, betöltés, szöveg módosítása, súgó.



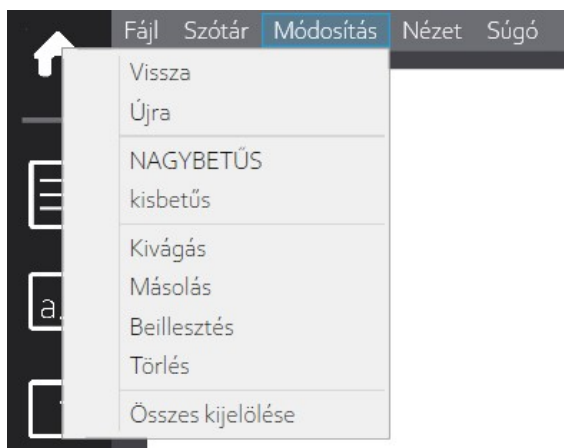
3.4. ábra. Fájl menüpont

A fájl menüpontban a fájlhoz tartozó felsorolt műveletek elvégzésére van lehetőségünk.



3.5. ábra. Szótár menüpont

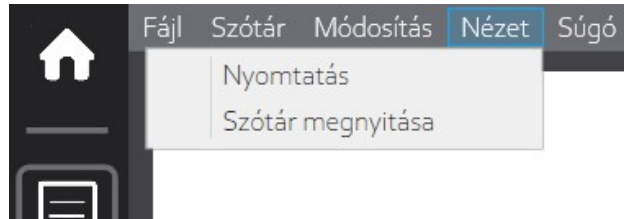
Az éppen szerkesztett dokumentumhoz lehetőségünk van szótárat betölteni, vagy az aktuálisan használt szótárunkat elmenteni új fájlba. A szótár megnyitása menüpontra való kattintással a jobb oldalon nyílik meg a szótár szerkesztéséhez tartozó felület.



3.6. ábra. Módosítás menüpont

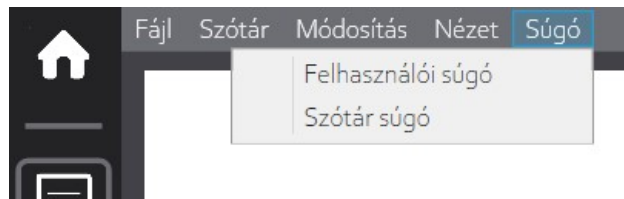
A papíron található szöveg módosítására szolgál ez a menüpont. Kijelölhetjük az összes, a papíron található szöveget (*ctrl+a*), törölhetjük a kijelölt szövegrészt (*delete*), kivágás (*ctrl+x*), másolás (*ctrl+c*), beillesztés (*ctrl+v*) műveleteket

végezhetünk el. Emellett a legutóbbi módosításainkat vissza is vonhatjuk (*ctrl+z*), a visszavont lépéseket újra alkalmazhatjuk(*ctrl+y*). Ezekre a műveletekre a zárójelekben megadott billentyűkombinációk is használhatók. Ezeken felül a kijelölt szöveget a megfelelő menüpontokra kattintva átállíthatjuk kis- és nagybetűsre is.



3.7. ábra. Nézet menüpont

A nyomtatás menüponttal megnyitjuk a nyomtatási beállításokat. Ennek a dialógusablaknak a segítségével van lehetőségünk a dokumentum elmentésére *pdf* fájlként. A szótár megnyitása menüpont a szótár kezelő felületét nyitja meg.



3.8. ábra. Súgó menüpont

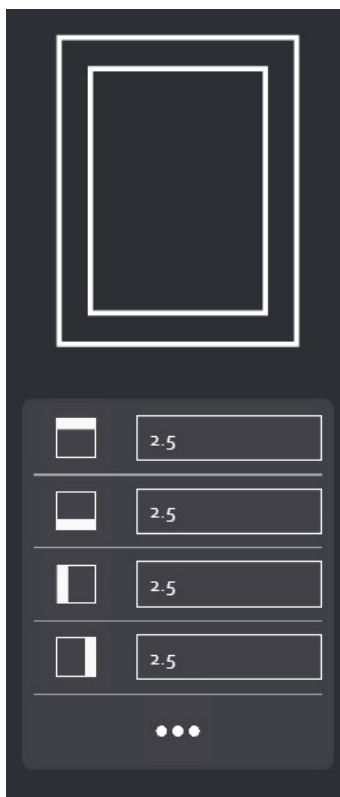
Az általános felhasználói segédletet és a szótár használatához tartozó segédletet is meg tudjuk nyitni a súgó menüpontból.

Az ablak jobb felső sarkában található három gombbal az alkalmazást letehetjük a tálcára (" \_ "), kinagyíthatjuk (" [ ] ") és bezárhatjuk ("X"). Az ablaknak két mérete közül tudunk választani, és a méreteket kinagyítás gombbal tudjuk változtatni. Nagyobb méretre váltáskor a jobb oldali felület nő meg, így a rajta található gombok és más elemek könnyebben használhatók, olvashatók lesznek azok számára is, akiknek nagyobb betűmérettel kell használniuk a számítógépet.

### 3.5.1. Oldalbeállítások menüpont

Az itt található szövegdobozok segítségével adhatjuk meg a dokumentumunkra alkalmazni kívánt margók értékeit. A szövegdobozban mindig az aktuálisan beállított érték szerepel. A beírt új értéket a mellette levő gombra kattintással tudjuk érvényesíteni. A gombok a következő sorrendben felelnek meg az oldalmargóknak: felső, alsó, bal, jobb.

Az ábra külön funkcióval nem rendelkezik, csak a margóbeállítás egyértelműsítésére szolgál.



3.9. ábra. Oldalbeállítások

A "..." gombra kattintva megnyitjuk a további oldalbeállítási lehetőségekhez tartozó párbeszédablakot.

Oldal beállítások

Margó beállítása

Felső 0,5

Bal 0,8

Jobb 0,4

Alsó 0,5

Alapértelmezett margó kiválasztása

Nagyon keskeny (0.5)

Keskeny (1.5)

Normál (2.5)

Vastag (3.5)

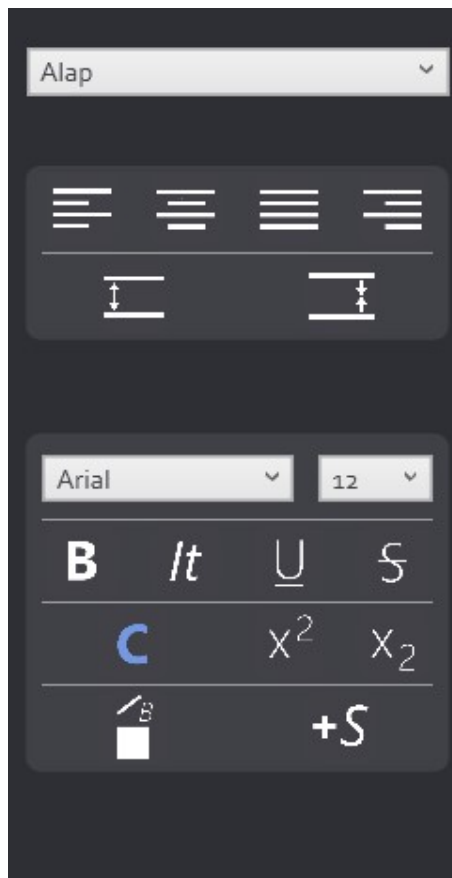
MÉGSEM OK

3.10. ábra. További lehetőségek

Az alsó listából előre elkészített margók közül választhatunk. Amint rájuk kattintunk, a párbeszédablak bezáródik, és megtörténik a kiválasztott margó beállítása a dokumentumra.

A nyilakra kattintva növelhetjük vagy csökkenthetjük az adott margót 0.1 centiméternek megfelelő értékkel. Az "OK" gombra kattintáskor az itt beállított értékek alkalmazódnak a dokumentumra, a párbeszédablak bezáródik. A mégsem gombra kattintva elvetjük a módosításokat, visszatérünk a dokumentumunkhoz.

### 3.5.2. Formázás menüpont



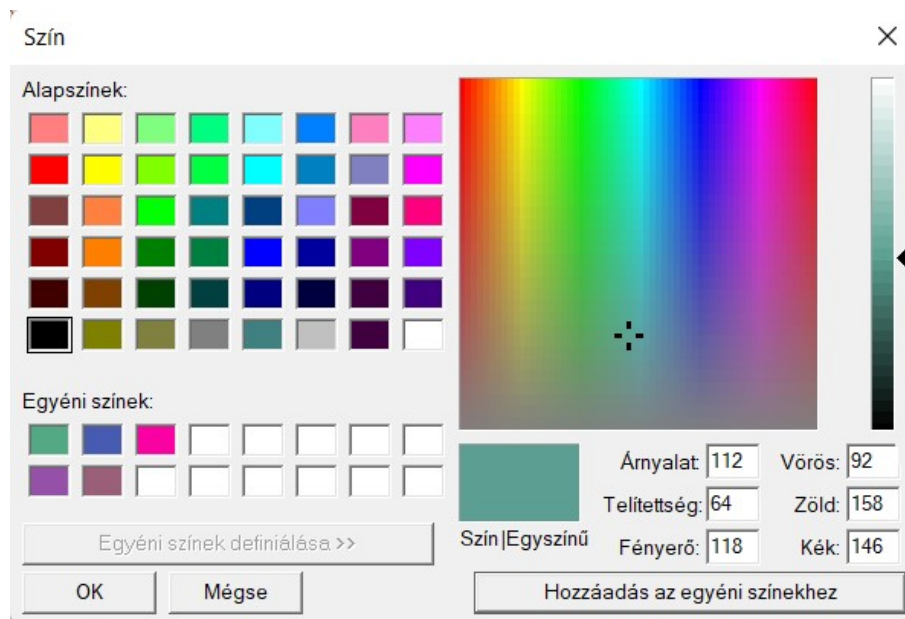
3.11. ábra. Szövegformázás felület

A legtöbb funkcióval rendelkező menüpont a formázásához nyújt lehetőségeket.

A felső lenyíló listából egy elemet kiválasztva a kijelölésre alkalmazhatunk az ott szereplő stílusok közül.

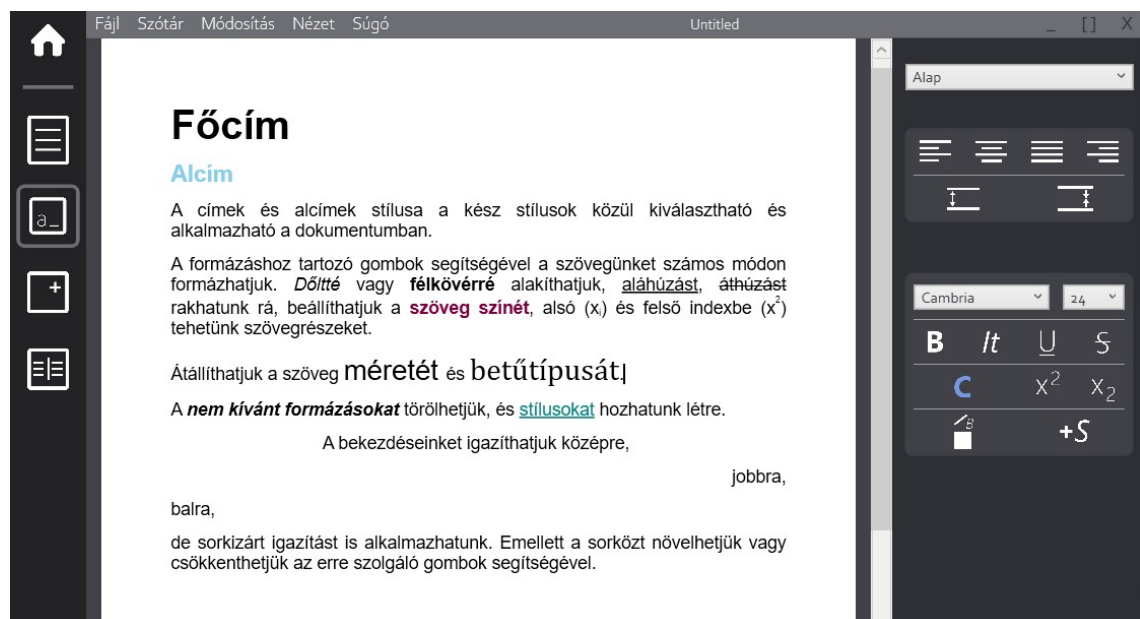
Az alatta található gombokkal a bekezdések igazítását állíthatjuk be. Balra, középre, sorkizártra és jobbra tudjuk igazítani a kijelölt, vagy az aktuális bekezdésünket. Az ez alatt található gombokkal a sorközt tudjuk növelni és csökkenteni.

A betűszín kiválasztásához egy párbeszédablak nyílik meg. Itt lehetőségünk van előre megadott színek közül választani, vagy saját színt keverni az erre kialakított felületen.



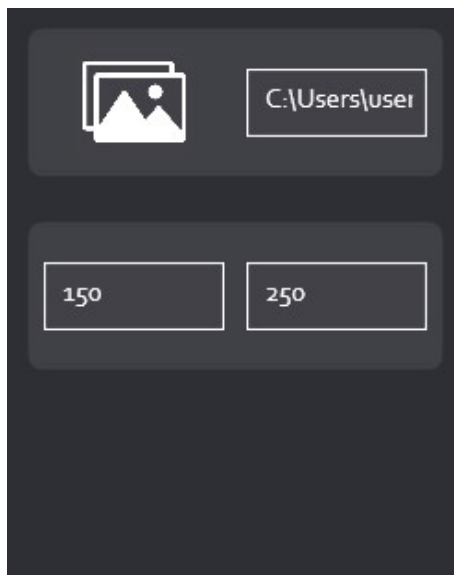
3.12. ábra. Ablak szín kiválasztásához

Emellett a kijelölt szöveget formázhatjuk még a felületen található gombok segítségével, felvehetünk új stílusokat is. Új stílus felvételekor a kijelölt szöveg legtöbb karakterére felvett formázási információkat mentjük el. Törölhetjük az alkalmazott formázásokat is, ilyenkor a dokumentumhoz tartozó alapvető betűszín, betűméret, betűtípusig csupaszítjuk le a kijelölt szövegrészt.



3.13. ábra. Szövegszerkesztési lehetőségek

### 3.5.3. Beszúrás menüpont



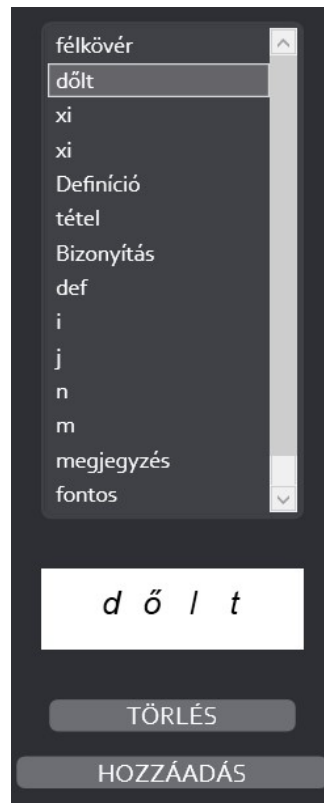
3.14. ábra. Képek beszúrása

Lehetőségünk van képeket is beszúrni a dokumentumunkba. A kép ikonra való kattintással előjön egy ablak, ahol kiválaszthatjuk a beszúrni kívánt képet. Az elérési útvonala a mellette levő szövegdobozba kerül.

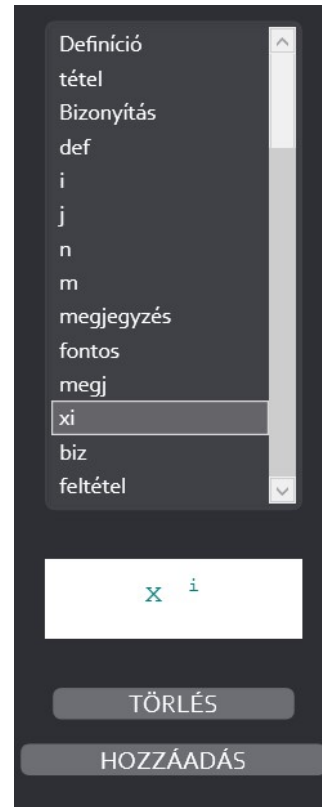
Az alsó két egymás melletti szövegdobozban a beszúrandó kép méreteit (szélesség, magasság) adhatjuk meg, ezt még a kép ikonra kattintás előtt be kell állítanunk. Ha nem adunk meg méretezést, akkor a kép alapértelmezett méretével kerül beszúrára, egy sor hosszának megfelelő maximális mérettel.



### 3.5.4. Szótár menüpont



3.15. ábra. Dőlt



3.16. ábra. Felső index

A szótárhoz tartozó felületen a szótárba felvett szavak listája, azok előnézetének megjelenítő felülete és a szótár műveleteinek gombjai találhatók.

A listából amikor kiválasztunk egy elemet, akkor alatta megjelenik a szóra eltárolt formázás előnézete. Ilyenkor lehetőségünk van a törlés gombbal eltávolítani a kiválasztott elemet. Ilyenkor a törölt elem a háttérben tárolt szótárunkból is eltávolításra kerül.

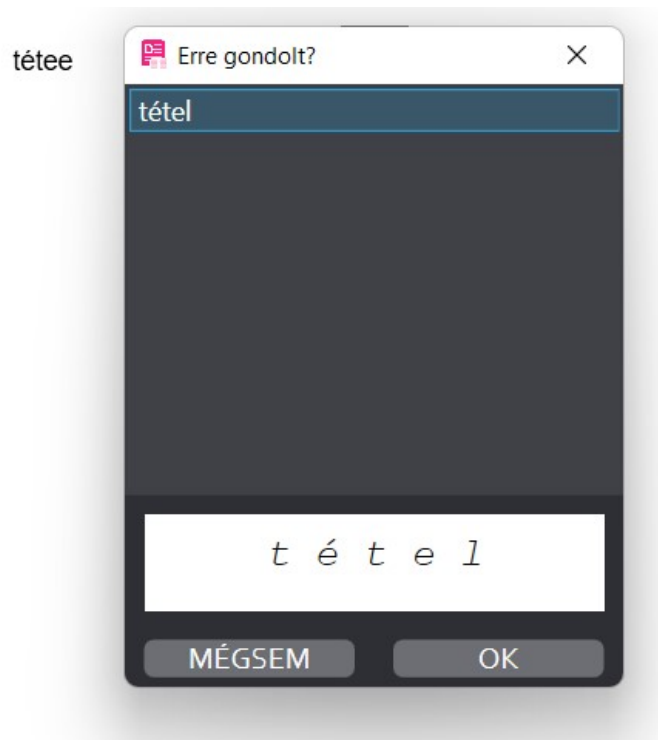
Lehetőségünk van a szótárhoz új elemeket is hozzáadni. Ehhez ki kell jelölnünk pontosan egy szót a dokumentumban, majd a hozzáadás gombra kattintva elmentjük a szótárba a kijelölt szót, és a rá alkalmazott formázásokat karakterről karakterre.

### 3.6. Szótár

A korábban taglalt szótár funkciók mellett az automatikus formázás működéséről és tudnivalóiról is érdemes néhány szót ejteni.

A szöveg begépelése közben, szóköz lenyomása után az utoljára beírt szóra az elemző alkalmazza a leggyakrabban használt elmentett formázást, amennyiben talált a szótárban a vizsgált szóval megegyező elemet. Ilyenkor a szótárban az elem gyakorisága eggyel nő. Ez azért fontos, mert egy adott szóhoz több különböző formázási stílust is fel tudunk venni a szótárba, de valahonnan tudnunk kell azt is a szótár elemeihez, hogy melyiket milyen valószínűséggel szeretné a felhasználó alkalmazni.

### 3.7. "Erre gondolt?"



3.17. ábra. A felugró ablak

Sok esetben hasznos lehet, és megkönnyítheti a begépelte hibák javítását az "Erre gondolt?" funkció. Minden begépelte, de a szótárban nem található szóra megvizsgál-

ja, hogy található-e a szótárban hozzá nagyon hasonló elmentett szó. Ha igen, akkor az első maximum három lehetőséget felajánlja nekünk, és választhatunk közülük egyet, vagy elutasíthatjuk az ajánlatot. Az ajánlat elutasításakor onnan folytathatjuk a dokumentum írását, ahol abbahagytuk. Ha kiválasztunk egy szót, akkor a szót, amit legutoljára begépettünk, kijavítja a kiválasztott szóra, és a hozzá elmentett formázást is alkalmazza rá a program.

### 3.7.1. Szótár létrehozása

Amikor üres dokumentumot kezdünk szerkeszteni, a szótár menüpontra kattintáskor először egy felugró ablakkal találkozunk, ha nincsen betöltött szótárunk. Itt megadhatjuk a szótár nevét, és az "OK" gombra kattintással létrehozzuk a *.json* fájlt, és betöltjük az üres szótárunkat, amennyiben helyes fájlnevet adtunk meg.

### 3.7.2. Szótár betöltése

A szótárt *.json* fájlként tároljuk, így megfelelő tartalmú *.json* fájlok betöltésére van lehetőségünk, amikor egy új szótárat szeretnénk betölteni az aktuális dokumentumunkhoz. A betöltés után az elemző az új szótárat fogja használni, az abban található formázásokat fogja automatikusan alkalmazni a begépett szavakra.

## 3.8. Mentés, betöltés

A szerkesztett dokumentumunkat a program futása során bármikor el tudjuk menteni. A főoldal bezárása előtt az alkalmazás egy felugró ablakban jelzi, ha mentés nélkül szeretnénk kilépni. Ilyenkor lehetőségünk van visszalépni, és kilépés előtt elmenteni az éppen szerkesztett fájlt.

Az alkalmazás indításakor betölthetünk korábbi fájlokat, és folytathatjuk azoknak a szerkesztését. Ilyenkor a hozzá tartozó szótárat is betöltjük a dokumentummal együtt.

A mentés és a betöltés is külön dialógus ablak segítségével történik.


A dokumentum (.xaml) és a szótár (.json) külön, önmagukban is használható fájlakként vannak elmentve, ezekhez tartozik még egy összekötő fájl, aminek a kiterjesztése .doced. Betöltéskor ezt a fájlt kell megkeresni. Amennyiben mentés után nem lettek a fájlok másik mappákba mozgatva és/vagy átnevezve, úgy mindhárom fájl betöltését biztosítja az összekötő fájl.

## 3.9. Egyéb információk

A program magyar nyelvű verzióval készült el. A felhasználói segédlet és a szótár súgó is magyar nyelven érhető el.

## 3.10. Gombok jegyzéke

Mivel sok gomb csak ikonnal jelzi a felhasználó számára, hogy mi a funkciója, így sok esetben intuitív használatot követelnek meg. Ezt hivatott megkönnyíteni az alábbi táblázat, ahol a gombok ikonjai és a funkciói fel vannak tüntetve.

|   |                           |
|---|---------------------------|
|  | Oldalbeállítások menüpont |
|  | Formázás menüpont         |
|  | Kép beillesztése menüpont |
|  | Szótár menüpont           |
|  | Kezdőoldal megnyitása     |
|  | Felső margó beállítása    |
|  | Alsó margó beállítása     |
|  | Bal margó beállítása      |
|  | Jobb margó beállítása     |

|           |  |
|-----------|--|
|           | Kép beillesztése                       |
|           | Balra zárt szövegigazítás beállítása   |
|           | Jobbra zárt szövegigazítás beállítása  |
|           | Középre zárt szövegigazítás beállítása |
|           | Sorkizárt szövegigazítás beállítása    |
|           | Sorköz növelése                        |
|           | Sorköz csökkentése                     |
| <b>B</b>  | Félkövér formázás beállítása           |
| <i>It</i> | Dőlt formázás beállítása               |
| <u>U</u>  | Szöveg aláhúzása                       |
| <u>S</u>  | Szöveg áthúzása                        |
| $x^2$     | Felső index beállítása                 |
| $x_2$     | Alsó index beállítása                  |
| <b>C</b>  | Szöveg színének beállítása             |
|           | Formázások törlése                     |
| <b>+S</b> | Új stílus felvétele                    |

3.1. táblázat. Az alkalmazás felületein található gombok és funkcióik

## 4. fejezet

# Fejlesztői dokumentáció

A fejlesztői dokumentáció a program specifikációját, felépítését, osztályszerkezetét, felhasználói felületét mutatja be. Emellett a fejlesztői és a tesztkörnyezetet is megismerhetjük. Az alkalmazás forráskódjának elvi felépítése és a program egyes részeiben alkalmazott megoldások leírása is megtalálható ebben a fejezetben.

### 4.1. Specifikáció

#### 4.1.1. Feladat

A feladat egy Windows operációs rendszeren futó asztali alkalmazás létrehozása, ami dokumentumok létrehozására, szerkesztésére használható. A szöveg szerkesztése során kezelni tudunk egy úgynevezett szótárat, ahol a korábban eltárolt formázások találhatók, amiket a program automatikusan alkalmaz a begépelte szavakra, amik már szerepelnek a szótárban, és alapszintű hibajavítást alkalmazva ajánl lehetséges formázásokat azokra a begépelte szavakra, amiket nem talál meg a szótárban.

A leírás alapján a program két fő egységre bontható. Az egyik maga a szövegszerkesztő alkalmazás, a másik pedig az elemző program, ami a szótárat kezeli.

### 4.2. A rendszer architektúrája

Az alkalmazás az objektum orientált (OOP) paradigma alapján és modell-nézet-nézetmodell (MVVM) architektúrával épül fel. A kód üzleti logikai részei a "model" projektben foglal helyet. A perzisztencia réteget külön modul biztosítja, minden ehhez tartozó kód a "persistence" projektben található. A "view" projektben szere-

pel az egyes ablakok, dialógusablakok és vezérlők leírása, ezekben az osztályokban és XAML fájlokban találhatók meg a megjelenítéshez tartozó kódok. A programot osztályokra bontottam, ezek jól körülhatárolt funkcióval rendelkező típusokat valósítanak meg. Minden osztály a funkciójának megfelelő adatokat és az adatokon elvégezhető műveleteket tartalmazza.

### 4.2.1. Fejlesztői környezet

A program fejlesztése Windows x64 architektúrájú rendszeren történt, Visual Studio fejlesztői környezetet használva, ami idén lett 25 éves.

### 4.2.2. A forráskód beszerzése

A program forráskódja megtalálható a dolgozathoz mellékelve, emellett letölthető a verziókezelőből is. Az alábbi URL címen (<https://github.com/benefruzs/bsc-thesis>) letölthető ZIP tömörített archívumként, vagy Git használatával a repository clone-ozásával is beszerezhető.

Az így megkapott mappa tartalmazza az összes forrásfájlt és állományt, amire szükségünk lehet a program fejlesztése során.

### 4.2.3. Könyvtárszerkezet

A beszerezhető mappa tartalmazza a program forráskódját, a hozzá tartozó teszt projektet, egy lefordított Windows x64 architektúrához megfelelő binárist, a dokumentációt és a verziókezeléshez szükséges fájlokat.

A projekt szerkezetét tekintve egy fő WPF .NET projektből és több WPF Class Libraryből áll. Ez a projektstruktúra biztosítja a könnyű verziókezelést és tesztelést. A program minden rétegéhez és különálló egységéhez külön projekt fájl tartozik.

A főkönyvtár tartalmaz a mappáktól külön álló fájlokat is. A README.md fájl a GitHub verziókezelőben megjelenő leírást tartalmazza a projektről. A .gitignore fájl tartalmazza azoknak a fájloknak a megnevezését, amit a Git verziókezelőnek figyelmen kívül kell hagynia.

#### 4.2.4. Névválasztási és kódolási konvenciók

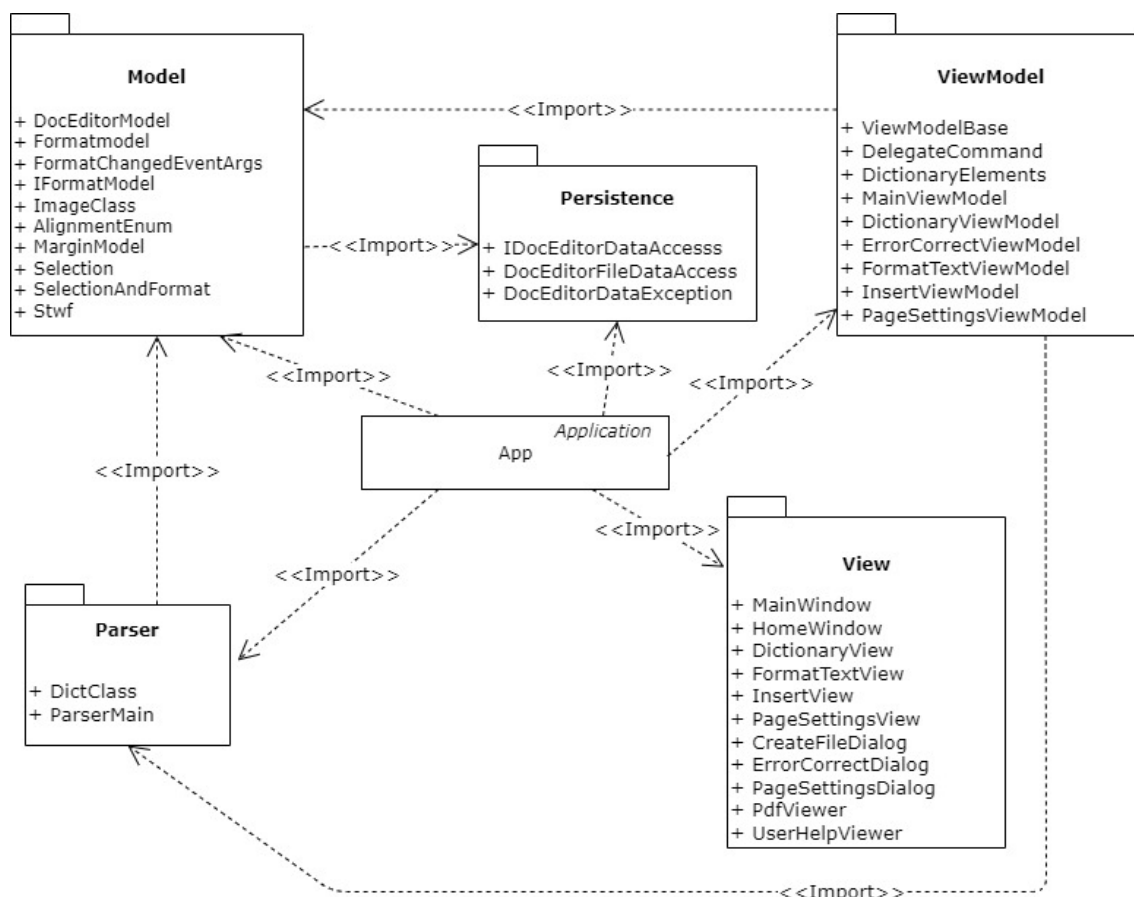
A programkód fontos összetevője az egységesen elnevezett azonosítók alkalmazása. A választott neveknek beszédesnek és konvencionálisnak kell lenniük. A jól megválasztott elnevezések nagyban hozzá tudnak járulni a programkód megértéséhez. A kódolásnál fontos még a Clean Code tulajdonságainak alkalmazása. Ennek követése biztosítja a kód olvashatóságát, karbantarthatóságát, tesztelhetőségét, és a megírt kódunk elegáns lesz ezáltal. Emellett a hibák könnyű felderítéséhez és a gyorsabb fejlesztéshez is hozzájárul. Névválasztási és kódolási konvenciók, amiket alkalmaztam:

- angol nyelvű elnevezések, kommentek,
- a forrásfájlok és osztályok nevei nagybetűvel kezdődnek, CamelCase jelölést alkalmazva,
- a publikus változók nevei nagybetűsek, a privát változók nevei kisbetűvel kezdődnek, CamelCase jelölést használva mindkét esetben,
- a metódusok, függvények nevei nagybetűvel kezdődnek, CamelCase jelöléssel,
- minden fájl pontosan egy osztályt tartalmaz,
- a függvényekhez tartozó kapcsos zárójelek mindig új sorban kezdődnek
- propertyk használata.

#### 4.2.5. Az alkalmazás struktúrája

A szoftver csomagdiagramja (package diagram) a 4.2. ábrán látható.





4.1. ábra. Az alkalmazás csomagdiagramja

### 4.3. Felhasználói felület

A jól áttekinthető, gondosan kidolgozott, könnyen kezelhető felhasználói felület alapvető követelmény minden szoftvernél.

Mivel asztali alkalmazásról van szó, ezért nem kell igazodni jelentősen eltérő méretű kijelzőkhöz. Az alkalmazás egy általánosságban bármilyen kijelzőméreten jól használható méretben nyílik meg, az ablak nem átméretezhető, de átállítható eggyel nagyobb méretre, amivel a jobb oldalon található oldalsávban elhelyezett gombok mérete növekszik. Ez a két adott méret biztosítja a legkényelmesebb alkalmazáshasználatot.

A felhasználói felület kódja elkülönül az alkalmazás többi kódjától. A felhasználói felülethez több ablak és dialógusablak tartozik, ezek XAML féligstrukturált nyelven vannak megírva.

### 4.3.1. Kezdőoldal

Az alkalmazás kezdőoldala az alkalmazás elindításához szükséges műveletek gombjait tartalmazza. A kezdőoldal a **HomeWindow**, ennek a megnyitásával indul az alkalmazás. A felületén gombok, egy lista és labelek találhatók. A felületre mentéshez, betöltéshez szükséges gombok vannak felvéve, és egy lista, aminek az elemeire kattintással elérhetünk kész szótár fájlokat, amiket betölthetünk új dokumentum létrehozáskor.

### 4.3.2. Az alkalmazás főoldala

A főképernyőt a **MainWindow** osztály implementálja, itt található a dokumentumok szerkesztéséhez egy **RichTextBox**, egy felső menüsor ahol a fájlhoz, a dokumentum szerkesztéséhez és a szótárhoz tartozó műveleteket láthatunk, emellett innen lehet megnyitni a súgókat. A jobb oldalon található sávra, a bal oldali menünek megfelelő User Control felületeket töltök be. Ezek programkódja külön fájlokban található, és mindegyikhez külön viewmodel tartozik. Mindig az éppen a menüből kiválasztott jelenik meg a főoldalon, ennek a funkcióit lehet használni. Az oldalbeállításokhoz tartozó User Control a **PageSettingsView**, a formázás menüponthoz tartozó a **FormaTextView**, a beszúráshoz az **InsertView** tartozik, a szótárhoz pedig a **DictionaryView**. Ezeknek a működését a hozzájuk hasonlóan elnevezett nézetmodell fájlok biztosítják.

### 4.3.3. Dialógus ablakok

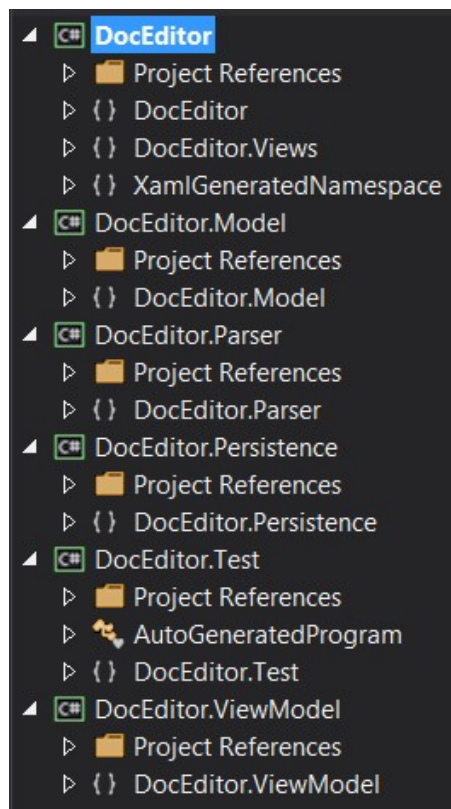
A dialógusablakok a felhasználóval való kommunikációra, hibák, pontatlanságok jelzése, lehetőségek felkínálása ("Erre gondolt?", "Kilép mentés nélkül?") szolgálnak. Az alkalmazásban használok előre elkészített és általam elkészített dialógus ablakokat is. A szótárhoz tartozó fájl létrehozásakor a **CreateFileDialog**, a margók további beállításainak megnyitáskor a **PageSettingsDialog**, a beírt szó javítására felajánlott lehetőségek megjelenítésekor a **ErrorCorrectDialog** osztály által implementált ablak jelenik meg. A súgó fájlok megtekintéséhez a **PdfViewer** és **UserHelpViewer** osztályok ad környezetet. A betűszín választásához a WinForms-ban beépített Color Picker Dialogot használok, ez az egyetlen WinForms-os elem, ami megjelenik a programomban.

## 4.4. Felhasználói súgó

A felhasználói súgót a *user-help.html* fájlként készítettem el. A weboldal stílusinformációi a *style.css* fájlban találhatók meg. A weboldal tartalma nagyban egyezik a dolgozat felhasználói dokumentációjával, felhasználóbarát formában tálalva.

## 4.5. Osztályszerkezet

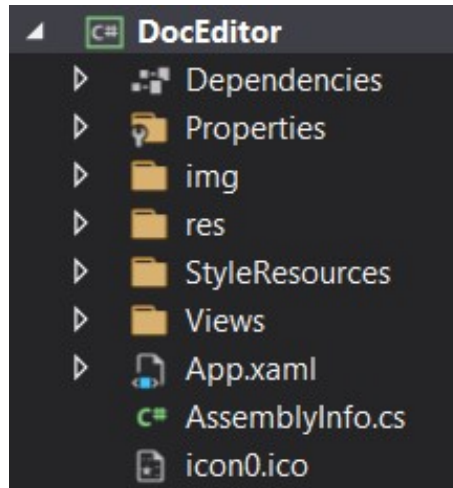
Az alkalmazás csomagdiagramja a 4.2. ábrán látható, rajta az összes osztályt feltüntetve. A program hat csomagot tartalmaz. Ezek az alkalmazás rétegeinek és funkcióiknak megfelelően vannak egymástól elkülönítve. A *DocEditor* felel a nézetért és a környezetért, a *ViewModel* tartalmazza a nézetmodell réteghez tartozó osztályokat, a *Persistence* a perzisztencia rétegért felelős, a *Model* a modell részt valósítja meg, a *Parser* pedig az elemző osztályaihoz tartozó csomag. Ezek mellett még egy teszt projekt is tartozik az alkalmazáshoz.



4.2. ábra. A forrásprojektek szerkezete

### 4.5.1. A nézet felépítése

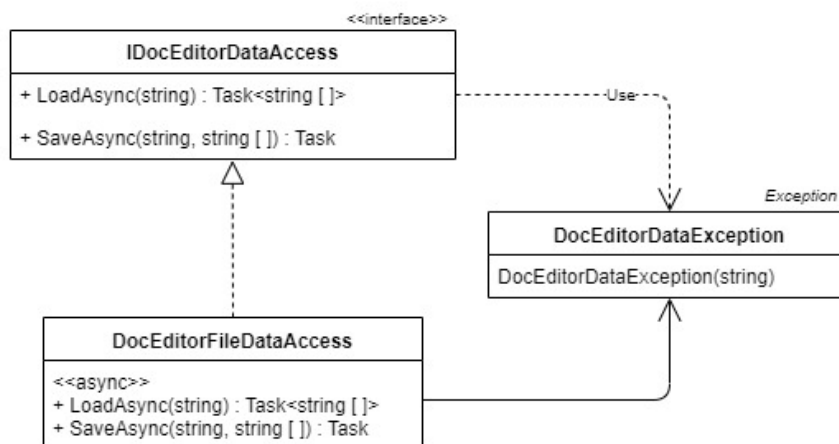
A program több külön ablakkal rendelkezik, ezek megnyitása a program aktuális állapotától függ. A nézet részletes leírása a felhasználói felület részben található.



4.3. ábra. A nézethez tartozó projekt

A grafikus felhasználói felületet leíró fájlok XAML nyelven készültek, a *Views* mappában találhatók. A felületi elemek stílusait külön fájlokban definiáltam, ezek a *StyleResources* mappában helyezkednek el, és az **App** teszi lehetővé az összekötésüket a nézet osztályaival. Az alkalmazás által felhasznált képi és más külső fájl erőforrások az *img* és *res* mappákban találhatók. Ikon fájlként (*icon0.ico*) a projekt része a program ikonja is.

## 4.5.2. Perzisztencia



4.4. ábra. A Persistence csomag osztálydiagramja

Az adatkezelési réteg feladata a modell fő információinak tárolása és a betöltés, mentés műveletek megvalósítása. Az **IDocEditorDataAccess** interfész nyújt lehetőséget a hosszútávú adattárolásra, a betöltésre (**LoadAsync**) és a mentésre (**SaveAsync**). Aszinkron módon valósítottam meg ezeket a műveleteket, mivel így hatékonyabb működést biztosítanak, amikor a változtatásokat nem egyenként, hanem egyszerre szeretnénk fájlba menteni. A **DocEditorFileDataAccess** osztály valósítja meg az interfészt szöveges fájl alapú adatkezelésre. A fájlkezelés során esetlegesen fellépő hibákat egy saját kivétellel jelzem, ez a **DocEditorDataException**.

Az alkalmazás háromféle fájlt tárol el. A szótárat tartalmazó *.json* fájlt, a dokumentum tartalmát *.xaml* formátumban, és az ezek összeköttetéséhez használt *.doced* fájlt. Ezzel a megoldással a szótár és a dokumentum fájljai önmagukban is használhatók, nem függenek egymástól. Betöltéshez az összekötő fájlt kell kiválasztani. Amennyiben egy mappában található a hozzá tartozó dokumentum és szótár fájllal, és azok neve nem változott meg a mentés óta, akkor az alkalmazásban be tudjuk tölteni mindhárom fájlt. Ezeket a hibalehetőségeket jól kezeli a program, hiba esetén érthető hibaüzenettel tér vissza a felhasználónak.

### 4.5.3. A .doced fájl tartalma

A fájl első sora tartalmazza a *.xaml* fájl elérési útvonalát, a második sora a *.json* fájlhoz tartozó útvonalat. A következő sorokban az oldalszélesség, oldalmagasság, oldalmargók, sorköz értéke és az elmentett stílusok listája szerepelnek.

### 4.5.4. A modell felépítése

A modell a szöveg szerkesztéséhez és az elemző működéséhez szükséges típusokat és műveleteket tartalmazza.



4.5. ábra. A Model csomag

A modell csomagjának a felépítése a 4.4. ábrán látható. Az üresen feltüntetett osztályok részletesebb diagramjai a későbbiekben következnek, itt csak a modell osztályai közötti kapcsolatok megjelenítéséhez szerepelnek.

A modell központi osztálya a **DocEditorModel**. Itt van nyilvántartva a dokumentum szerkesztéséhez szükséges összes információ, itt történik a teljes dokumentumra vonatkozó információk, mint az oldalmargók (**\_margin**), oldalméret (b>PageWidth, **PageHeight**), sorköz (**LineHeight**), kijelölt szöveg (**Select**), alapértelmezett formázási stílus (**\_formatText**) modell szinten történő kezelése. Itt található az összekötő fájl neve és elérési útvonala (**\_fileName**, **\_filePath**).

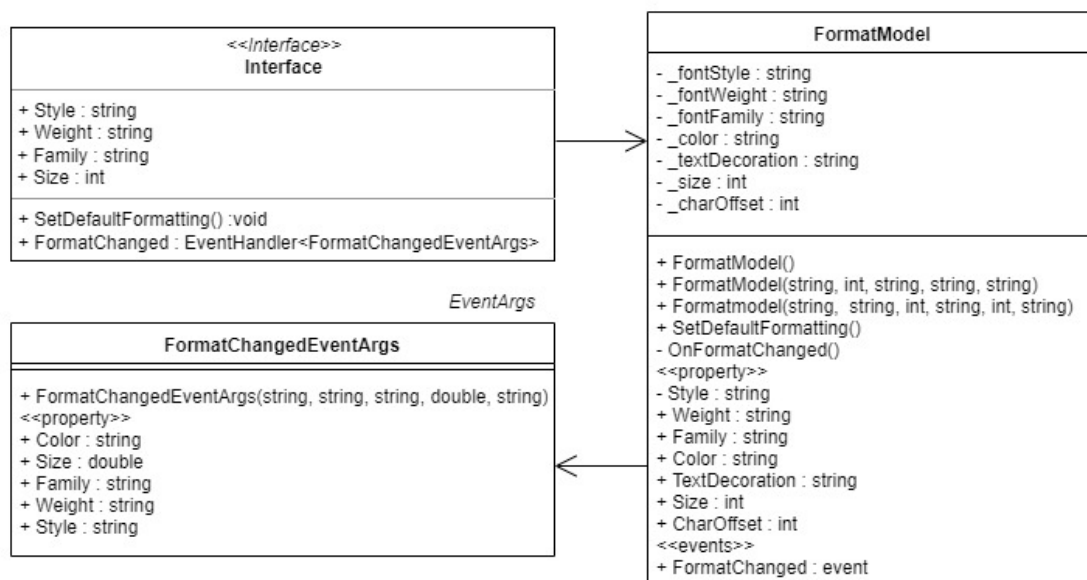
A bekezdések igazítását az **Alignment** felsorolási típuson keresztül kezeli a modell.

A modell példányosításakor beállítódnak az alapértelmezett értékek a margókra, sormagasságra, formázásra, megtörténik a szükséges adattagok inicializálása. Az osztály műveletei biztosítják többek között az oldalmargók ellenőrzött beállítását, erre a **SetTopMargin()**, **SetBottomMargin()**, **SetLeftMargin()**, **SetRightMargin()** és **SetAllMargins()** műveletek használatosak. Az ellenőrzésnél fontos, hogy ne lehessen negatív vagy túl nagy margókat beállítani. Az oldalmargók lekérdezéséhez adatkonvertáló műveleteket használnak, ezek a **GetTopMargin()**, **GetBottomMargin()**, **GetLeftMargin()**, **GetRightMargin()**. Ezen műveletek segítségével olyan formában lehet lekérdezni a az oldalmargókat, amivel a felhasználói felületen be lehet állítani azokat. A sorköz ellenőrzött növelése és csökkentése a **LineHeightIncr()** és **LineHeightDecr()** műveletekkel történik. Itt is fontos, hogy ne lehessen negatív vagy túl nagy sorközt beállítani.

Az alkalmazható előre elkészített kész stílusok listáját is a modell tárolja és kezeli, ez a **\_fontStyles**. A listához új elem hozzáadására az **AddNewFormatStyle()** művelet szolgál.

Példányosításakor megkapja még az adatkezelés felületét is, amelynek segítségével az összekötő fájl betöltésére (**LoadFileAsync()**) és mentésére (**SaveFileAsync()**) ad lehetőséget, aszinkron műveletekkel.

Fájl betöltésének eseményéről a **FileLoaded** és mentésének eseményéről a **FileSaved** események tájékoztatnak.



4.6. ábra. A formázásért felelős osztályok

Logikailag minden karakter, ami szerepel egy dokumentumban rendelkezik minden lehetséges stílusinformációval. Ennek a modellben történő ábrázolásához készítettem el a következő osztályt.

A szöveg formázását implementáló osztály a **FormatModel**, ami az **IFormatModel** interfészt valósítja meg. Az osztály adattagjai egy szövegre megadható formázásokat ábrázolnak. Az adattagok a betűstílusnak (`_fontStyle`, `_fontWeight`), betűtípusnak (`_fontFamily`), betűméretnek (`_size`), betűszínnek (`_color`), szövegdekorációnak (`_textDecoration`) és a felső- és alsó index igazításának (`_charOffset`) feleltethetők meg.

Az osztály biztosítja az alapértelmezett stílus létrehozását is (`SetDefaultFormatting()`). A formázás változásának jelzéséhez eseményt tudunk kiváltani az osztályban.

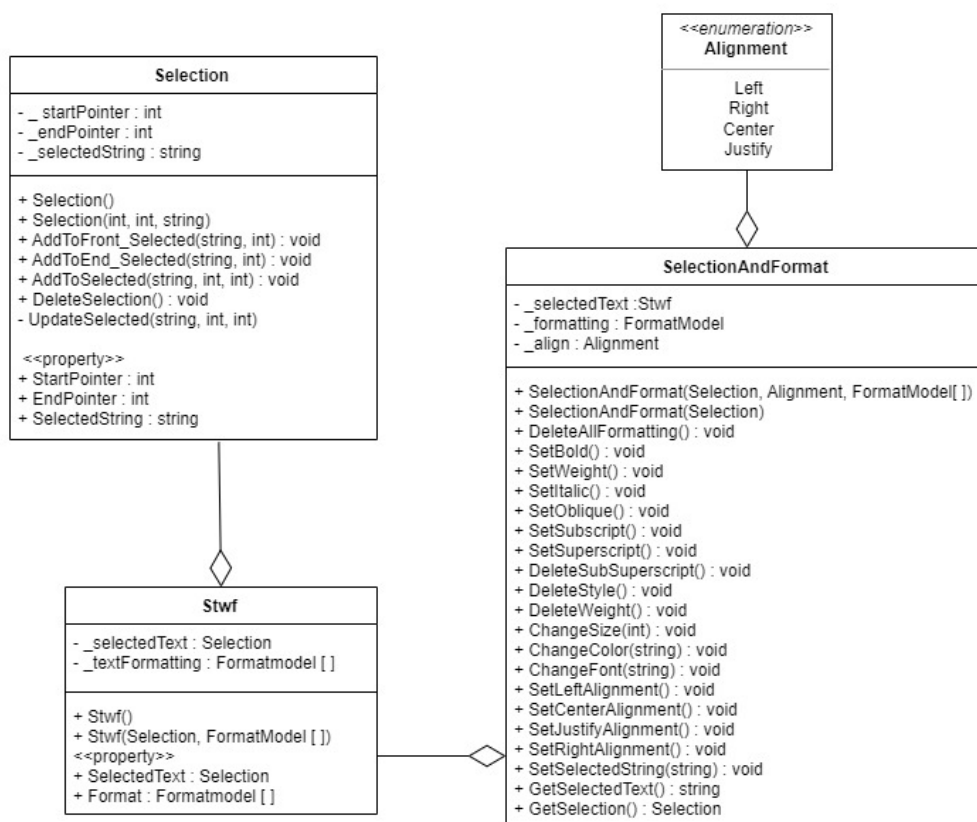
Minden szövegformázással kapcsolatos művelet alapjául ez az osztály szolgál. A fejlesztés során szinte elsőként készült el, mivel már kezdetben is egyértelmű volt,



hogy a szövegek szerkesztéséhez és az elemző működéséhez is szükség lesz rá. Az implementációját úgy készítettem el, hogy kényelmesen tudjam használni a felhasználói felületen való megjelenítéshez is, ezért a nem szám értékeket felvehető adattagok szöveg típusúak, viszont csak olyan értékeket vesznek fel, amit a nézetben található vezérlők is értelmezni tudnak. Úgy gondoltam, hogy ez kényelmes és jó módja ezeknek az információnak a kezelésének.

Az alsó és felső indexhez tartozó adattag szám típusú, és három lehetséges értéket vehet fel. Az 1 érték az alap állapot, -2 jelzi az alsó indexet, 2 a felső indexet. Ez jól használható a betűméret beállításához is, mivel alsó és felső indexben, ha felezem a betűméretet, akkor az a megjelenítésen pont jól fog szerepelni.

Mivel minden szövegrészre szükséges, hogy le tudjuk kérdezni és be tudjuk állítani a hozzá tartozó formázási információkat, a következő, a szövegszerkesztés és az elemző működésének megvalósításához elengedhetetlen osztályok mindegyikében szerepel ilyen típusú adattag.



4.7. ábra. A kijelölésért felelős osztályok

A dokumentumban kijelölt szöveget a modell rétegben a **Selection** osztály valósítja meg. Az adattagjai a kijelölt szöveget (**\_\_selectedString**), a kijelölés kezdő (**\_\_startPointer**) és végpozíciójának (**\_\_endPointer**) mutatóját tárolják. Az osztály konstruktorai biztosítanak üres osztálypéldány, és értékekkel feltöltött példány létrehozását is. A metódusai a kijelölt szöveg bővítésére (**AddToSelected()**) és törlésére (**DeleteSelection()**) szolgálnak.

Az **Stwf** (Selected Text With Formatting) osztály egy **Selection** osztálypéldányt, és a kijelölt szöveg karaktereire meghatározott formázási információkat kezeli. Ezt az osztályt használja az elemző a modell osztályai közül. Az elemzőnek átadott példány így pontosan tartalmazza magát a szót, a helyét az eredeti szövegben és a rá vonatkozó minden formázási információt. Ugyanennek az osztálynak egy példányával tér vissza az elemző a megfelelő műveletek elvégzése után, ami automatikus formázás esetén tartalmazza a formázandó szót, és a formázást.

Az éppen formázott szövegrészt még bővebben kezelő osztály a **SelectionAndFormat**. Egy **Stwf** típusú adattagban tárolja a kijelölt szót, emellett még a teljes kijelölésre vonatkozó formázási információkat (**\_\_formatting**) és az igazítást (**\_\_align**) is tartalmazza. A műveletei biztosítanak minden szövegformázás és szövegigazítás beállítását, törlését és lekérdezését.

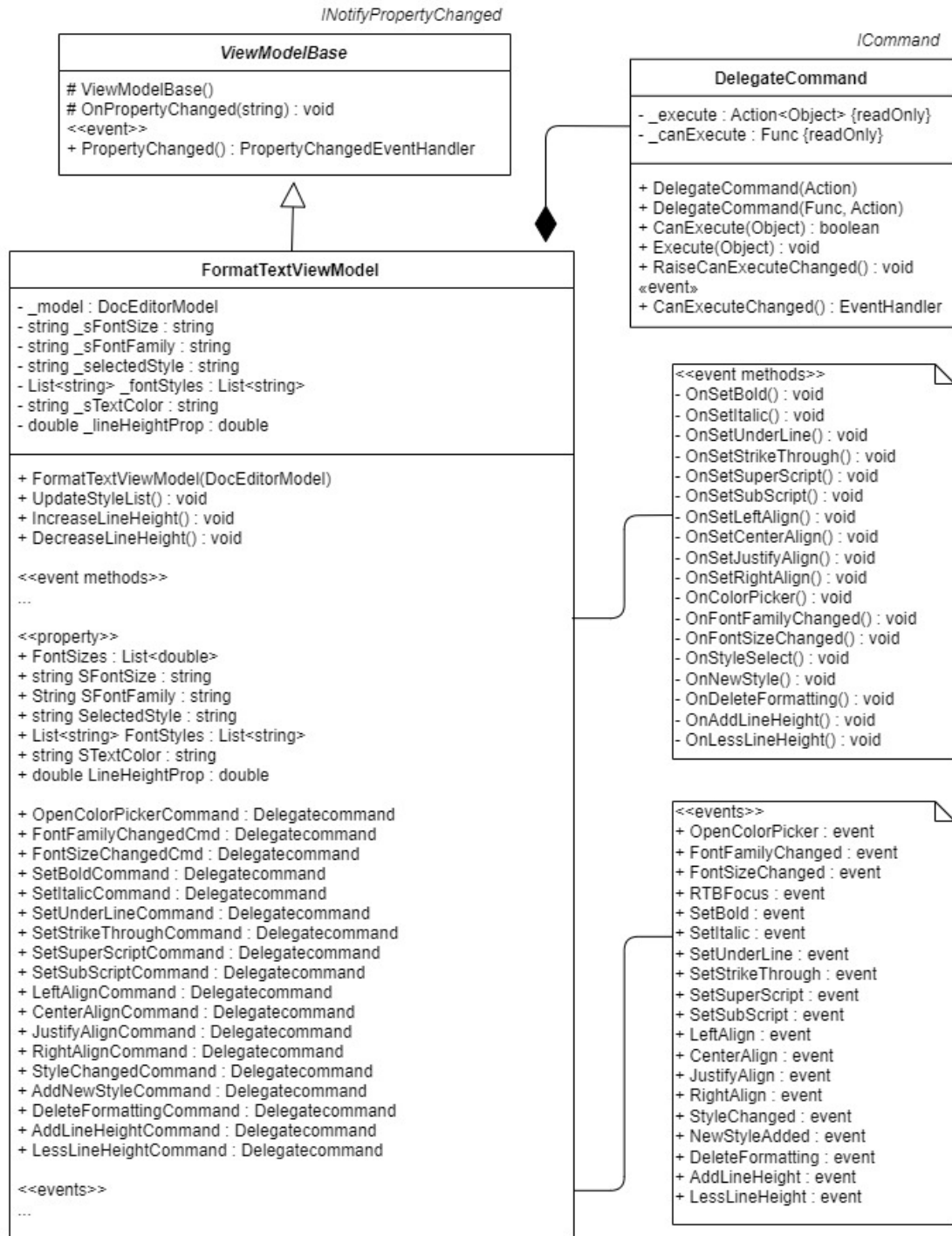
#### 4.5.5. A nézetmodell felépítése

A nézetmodell réteg a nézet és a modell közötti kommunikációt segíti elő. A nézetmodell több különböző funkcióval rendelkező önálló osztállyal van megvalósítva, ezek egymás közötti, és a modellel történő kommunikációja biztosítja a felület és a modell együttes működését. Minden osztály megvalósításához közös változásjelző ősosztályt (**ViewModelBase**) használok valamint egy általános utasítás osztályt (**DelegateCommand**) használtam. A parancsokhoz eseményeket kötöttem, amelyek a parancs lefutását jelzik a vezérlőnek. A nézetmodell minden osztálya tárolja a modell (**\_\_model**) és az elemző (**\_\_parser**) egy hivatkozását, amikből információkat kér le tőlük. Direkt nem avatkozik az alkalmazás futtatásába.



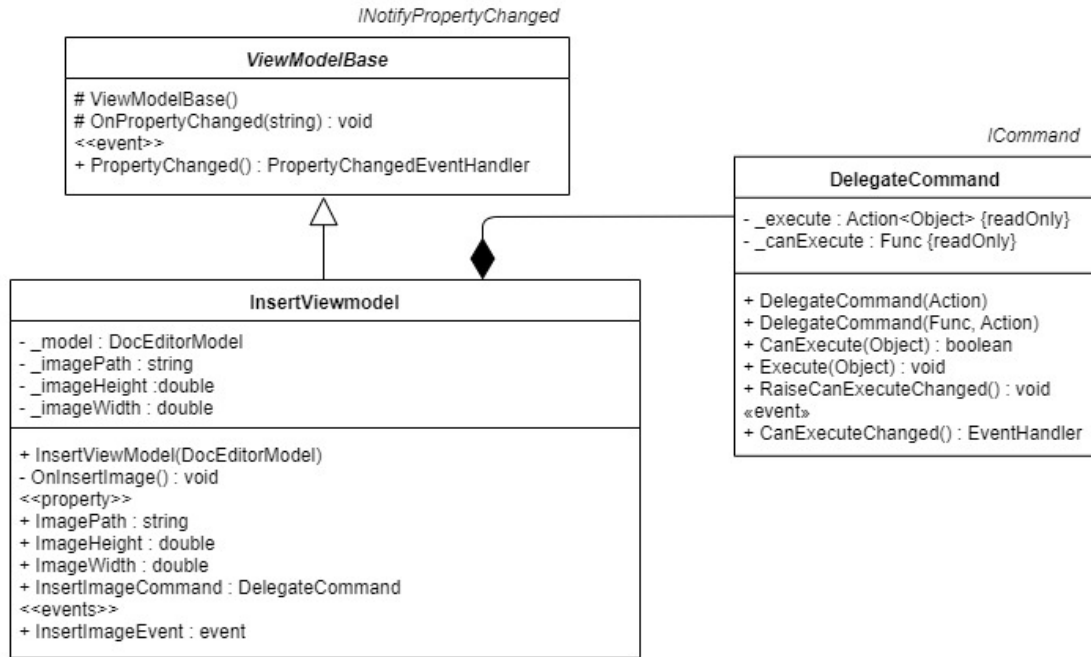
Mivel ez az osztály kommunikál a dokumentumot megjelenítő vezérlővel, így a nézetmodell réteg többi osztályának információit át kell tudnia venni. Ezt az **App** osztályon keresztül tudtam megvalósítani, így a környezetnek nagyon fontos szerepe van az alkalmazásom működésében.

42



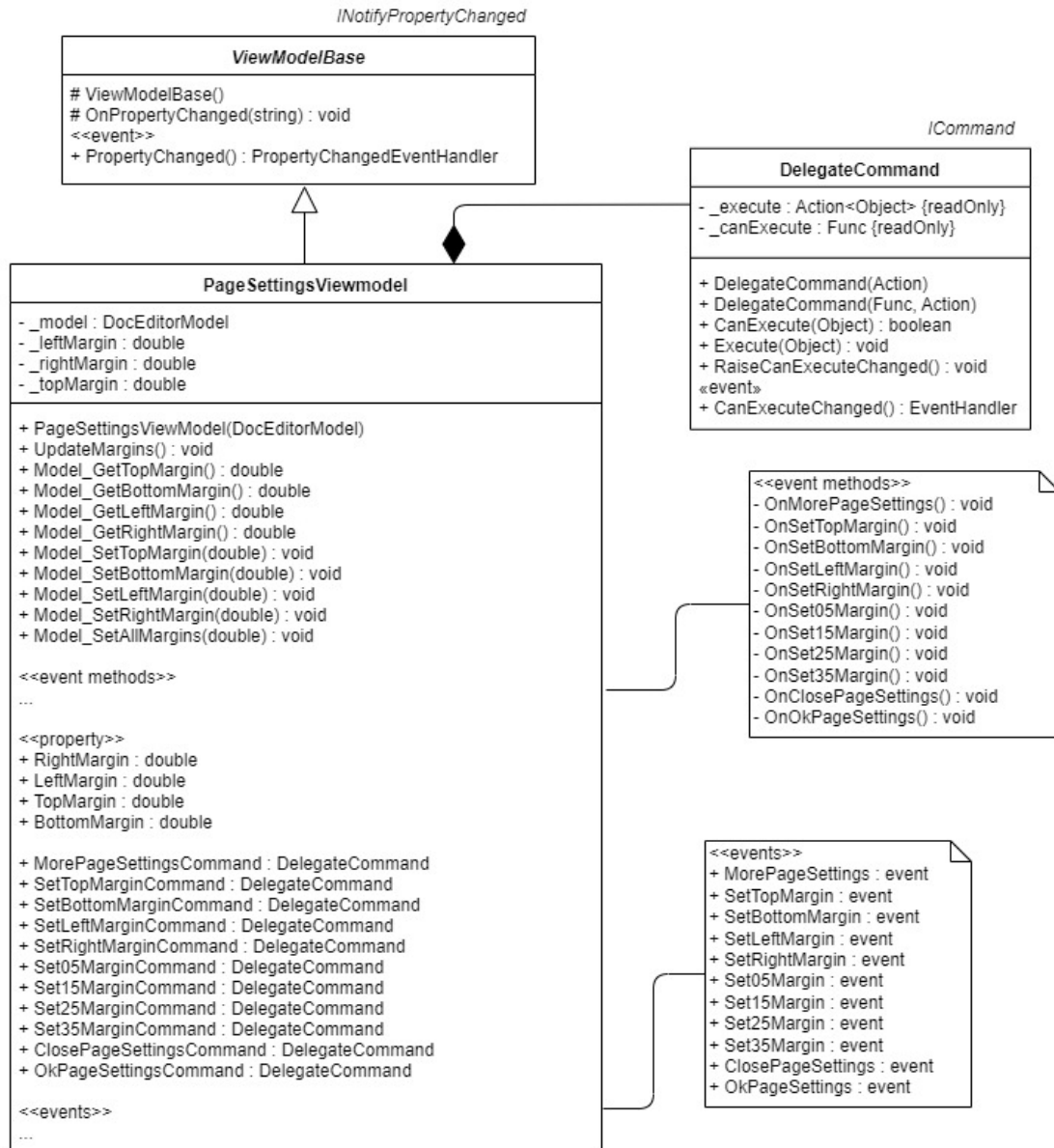
4.9. ábra. A szövegek szerkesztéséhez tartozó nézetmodell osztálydiagramja

A formázáshoz tartozó felület mögötti nézetmodell a **FormatTextViewModel** osztály implementálja. A felületen szereplő gombok, legördülő listák működését teszi lehetővé. A dokumentumban kijelölt szöveg formázásának, a bekezdések igazításának, a sorköz növelésének, csökkentésének, új stílusok felvételének parancsait jelzi és kezeli.



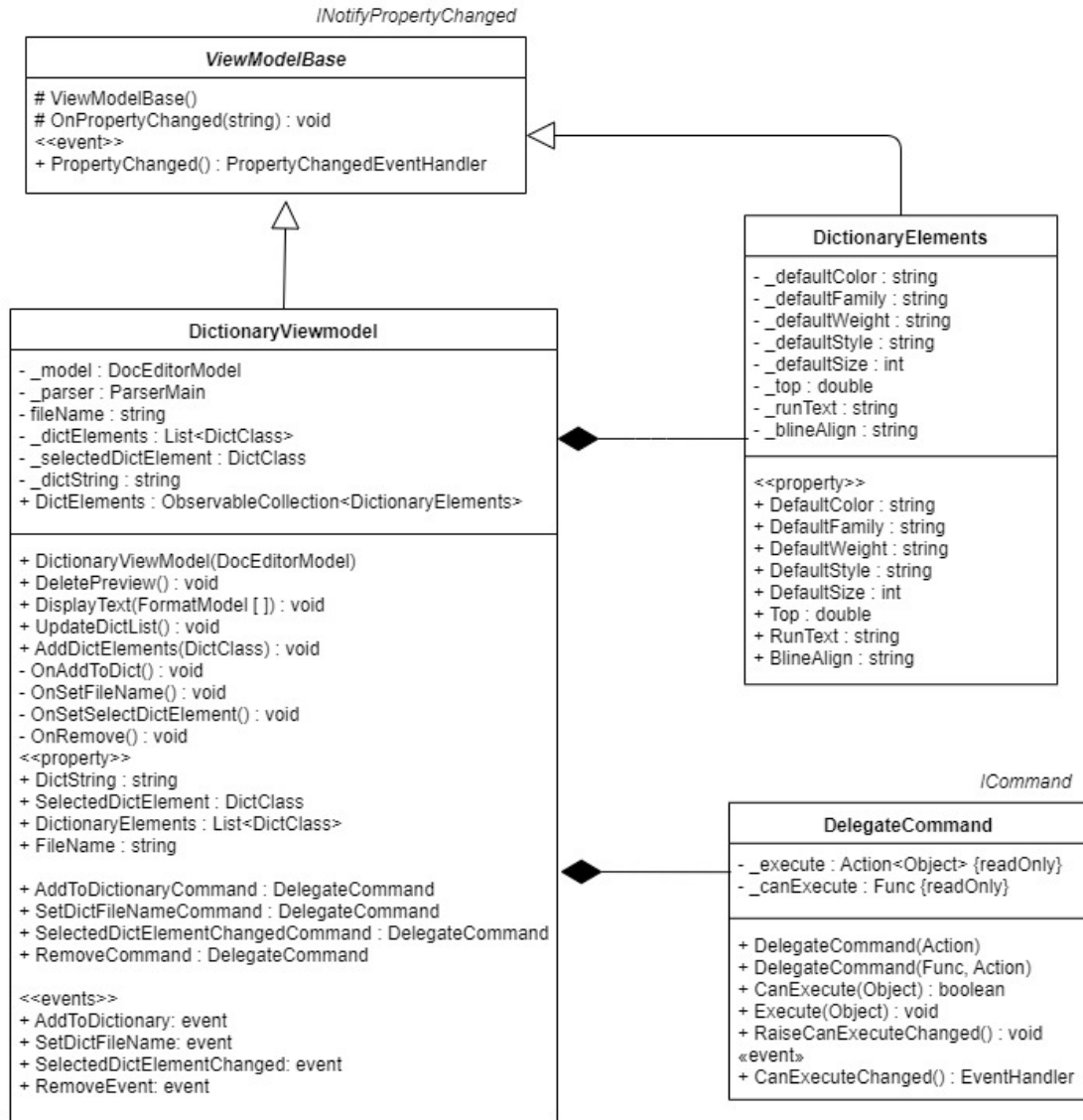
4.10. ábra. A képek beszúrásához tartozó nézetmodell osztálydiagramja

A dokumentumba beszúrt képek modellnek történő jelzéséért és a felhasználói felületen a kép beszúrásának működéséért felelős osztály az **InsertViewmodel**. A beszúrt kép elérési útvonalát (**\_imagePath**) és méretét (**\_imageHeight**, **\_imageWidth**) tárolja, és küldi a modellnek.



4.11. ábra. Az oldalbeállításokhoz tartozó nézetmodell osztálydiagramja

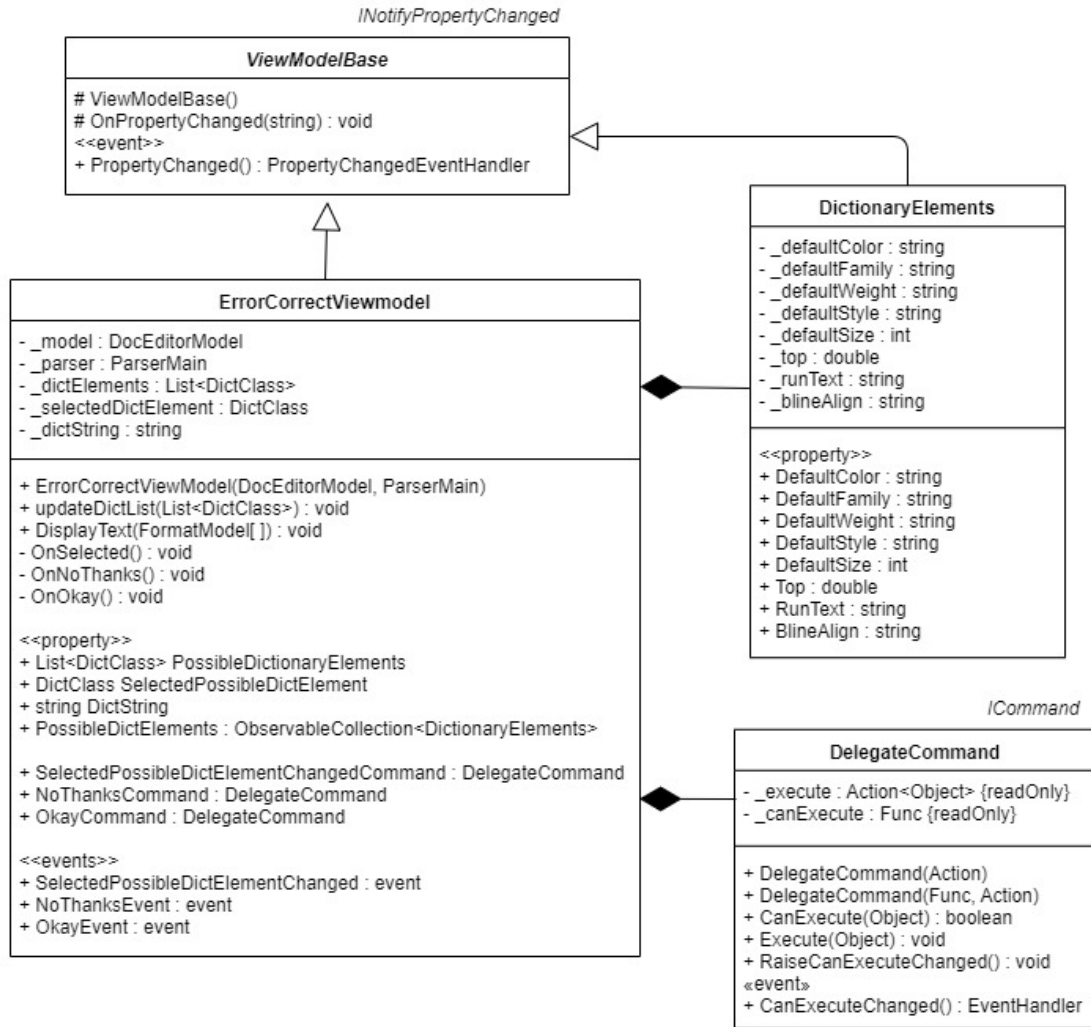
A **PageSettingsViewModel** az oldalmargók beállításáért felelős privát adat-tagok és publikus propertyk kezeléséért felel. A margók értékeit a modell margóként felelős adattagjai és műveletei alapján állítja be, illetve, ha a nézetben megváltoznak ezek az értékek, akkor a modellt is értesíti a frissítés esedékességéről. Ezeken kívül a **PageSettingsDialog** dialógusablak megnyitását és működését is irányítja.



4.12. ábra. A szótárhoz tartozó nézetmodell osztálydiagramja

A szótár elemeinek listáját, azok előnézetét és a szótárból való törléshez, a szótárhoz való új elem hozzáadásához tartozó gombok funkcióit valósítja meg az osztály.

A szótárhoz tartozó formázási stílusok előnézetének megjelenítéséhez egy külön osztályt biztosítok (**DictionaryElements**), ami eltárolja a megjelenítendő szöveget, a formázási információkat (szín, betűtípus, betűstílus, betűméret, igazítás). Ezeket az elemeket egy felügyelt gyűjteménybe (*ObservableCollection*) helyezem el a **DictionaryViewModel** osztályban. Ez biztosítja az előnézet dinamikus megjelenítését a felhasználói felületen.



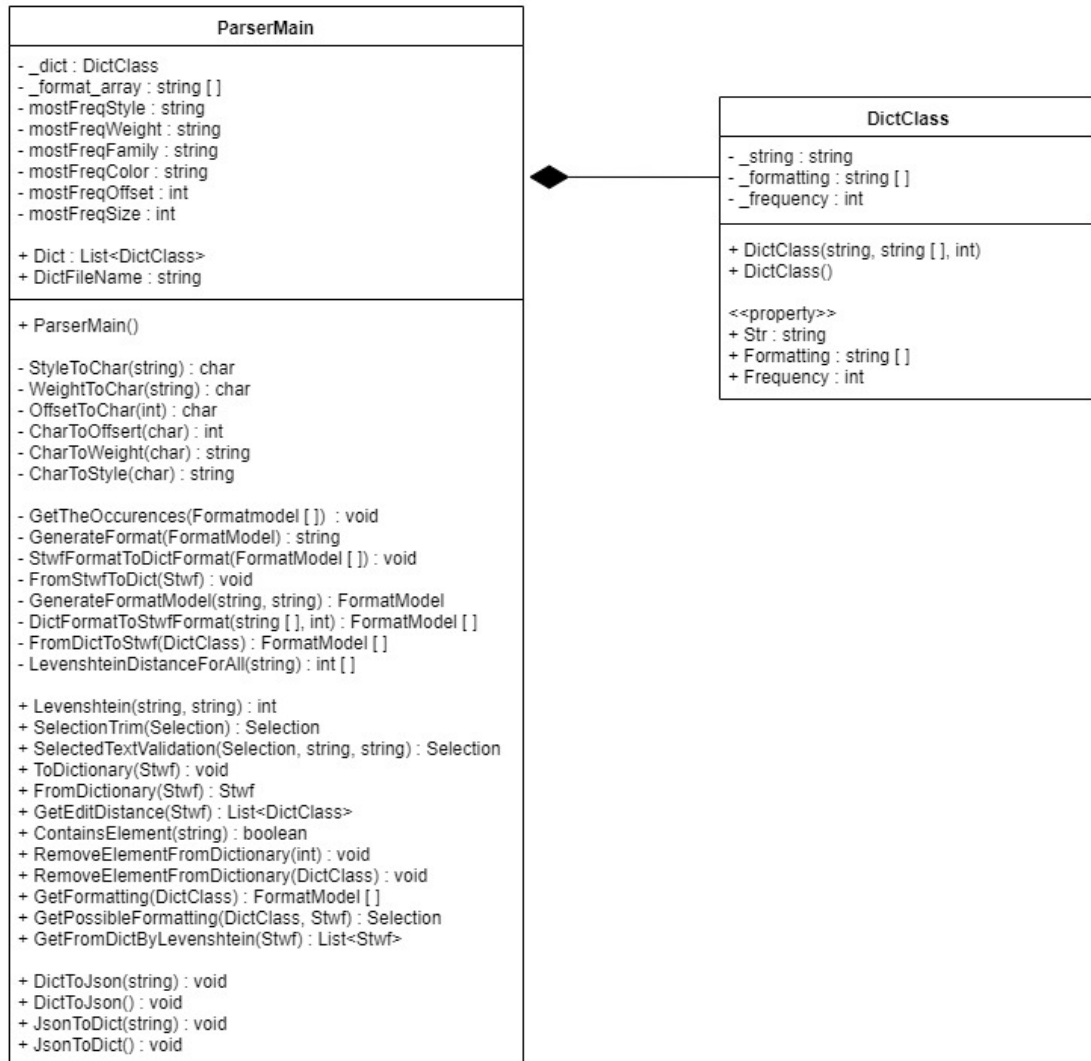
4.13. ábra. Az "Erre gondolt?" funkcióhoz tartozó nézetmodell osztálydiagramja

Az **ErrorCorrectViewmodel** osztály is használja a formázási stílusok előnézetének megjelenítéséhez tartozó **DictionaryElements** osztályt. Az elemeket itt is egy felügyelt gyűjteményben tárolom el. Emellett a lehetőségként felkínált szavak listájának kezeléséért felel.

#### 4.5.6. Az elemző felépítése

A szótárhoz tartozó elemző program két osztályt tartalmaz, egyik az elemző működéséért felelő osztály, ez lényegében az elemző "motorja", a másik osztály pedig a szótár elemeinek típusát adja meg.





4.14. ábra. Az elemző osztálydiagramja

A csomaghoz tartozó **DictClass** implementálja a szótár elemeinek típusát. Az adattagok a szótárba felvett szóból (**\_string**), a szó karaktereire meghatározott formázási információk tömbjéből (**\_formatting**) és az adott szó előfordulásának gyakoriságából (**\_frequency**) állnak. Az osztály lehetőséget biztosít üres osztálypéldány és értékekkel ellátott osztálypéldány létrehozására is.

Az elemző másik osztálya a **ParserMain**. Ez felel az elemző működéséért. Nyilvántart egy **DictClass** példányokból álló listát (**Dict**), ami a szótárba felvett szavakat és a hozzájuk tartozó információkat tartalmazza. Emellett egy privát **\_dict** példányt is kezel a **DictClass** osztályból, ami a listához hozzáadott új elem előzetes feldolgozásához van fenntartva. Az előzetes feldolgozáshoz egy tömböt is használ, amiben a szó formázási információinak már a szótárbeli formára átala-

kított alakját tárolja. Emellett az éppen betöltött szótár fájl elérési útvonalát is nyilvántartom az osztályban. Ez fontos ahhoz, hogy az elemzőt használó alkalmazás tudja, hogy van-e éppen betöltött szótár fájl, és, hogy melyik az.

Az osztály példányosításakor egy üres szótárat, és egy üres szótár objektumot inicializálok.

Az osztályban megtalálhatók adatkonvertáló műveletek, amik a kapott stílusinformációkat alakítják át a szótár számára a megfelelő karakterre, vagy a szótárban tárolt információkat alakítják vissza alkalmazható formára. Ezeket a műveleteket használják fel az osztály további átalakító és elemző metódusai. A **GetTheOccurrences()** művelet meghatározza az elemzőhöz kerülő szó olyan formázásait, amik a legtöbb karakterre érvényesek. A szótárban ez minden szóra a formázási tömb első elemében van feltüntetve. A tömb többi elemét a **GenerateFormat()** művelettel állítja be az elemző az átalakító segédműveletek segítségével.

Az elemzőhöz kerülő szó először ellenőrző műveleteken halad át. Erre szolgál a **SelectedTextValidation()** metódus, ami megvizsgálja, hogy pontosan egy értelmes szót szeretnénk-e hozzáadni a szótárhoz. Ez azért fontos, mert alapvető követelmény, hogy a szótárban csak értelmes, egy szóból álló elemek szerepeljenek. Ha a szótárhoz hozzáadni kívánt szó tartalmaz szóközt, számokat vagy speciális karaktereket, akkor jelzi a metódus meghívójának, hogy sikertelen volt a validálás. A szótárba új szót felvenni egy szó egy dokumentumban történő kijelölésével tudunk. A kijelölésről nem szükséges azt feltételezni, hogy az elején és a végén nem szerepelnek szóközők, viszont a szótár csak a szóközők nélküli szót tartalmazhatja, ezért az ellenőrzés előtt ezeket a szóközőket el kell távolítani. Erre a **SelectionTrim()** műveletet lehet alkalmazni.

Az elemző egyik fő művelete a **ToDictionary()**, ami a szótárba veszi fel új elemként a paraméterként átadott **Stwf** típusú elemet. Először a **\_\_dict** változóba átalakítja a szótárbeli szavak alakjára, majd ellenőrzi, hogy szerepel-e a szótárban már ugyanilyen elem. Ha igen, akkor csak a gyakoriságát növeli. Ha nem, akkor hozzáadja a szótárhoz az új elemet.

Szótárból szavakhoz tartozó formázás lekérdezésére **FormDictionary()** művelet szolgál. Először megvizsgálja a függvény, hogy szerepel-e a kapott szó a szótárban, ha igen, kiválasztja a leggyakoribb elemet, és átalakítja a szó jelenlegi formázását a szótárból vett információ alapján, az elem gyakoriságát növeli a szótárban. Ha nem találja meg a szót, akkor az elemző hibajavító algoritmusait tudjuk alkalmazni.

Az elemző Levenshtein távolságot számítva meghatározza a vizsgált szó távolságát a szótár összes szavától, ezt a **LevenshteinDistanceForAll()** művelet végzi el. Ez alapján készít egy listát, amiben az első három olyan elem szerepel a szótárból, aminek a szóval vett Levenshtein távolsága 2 vagy 1. Erre a **GetEditDistance()** művelet használatos.

#### 4.1. forráskód:

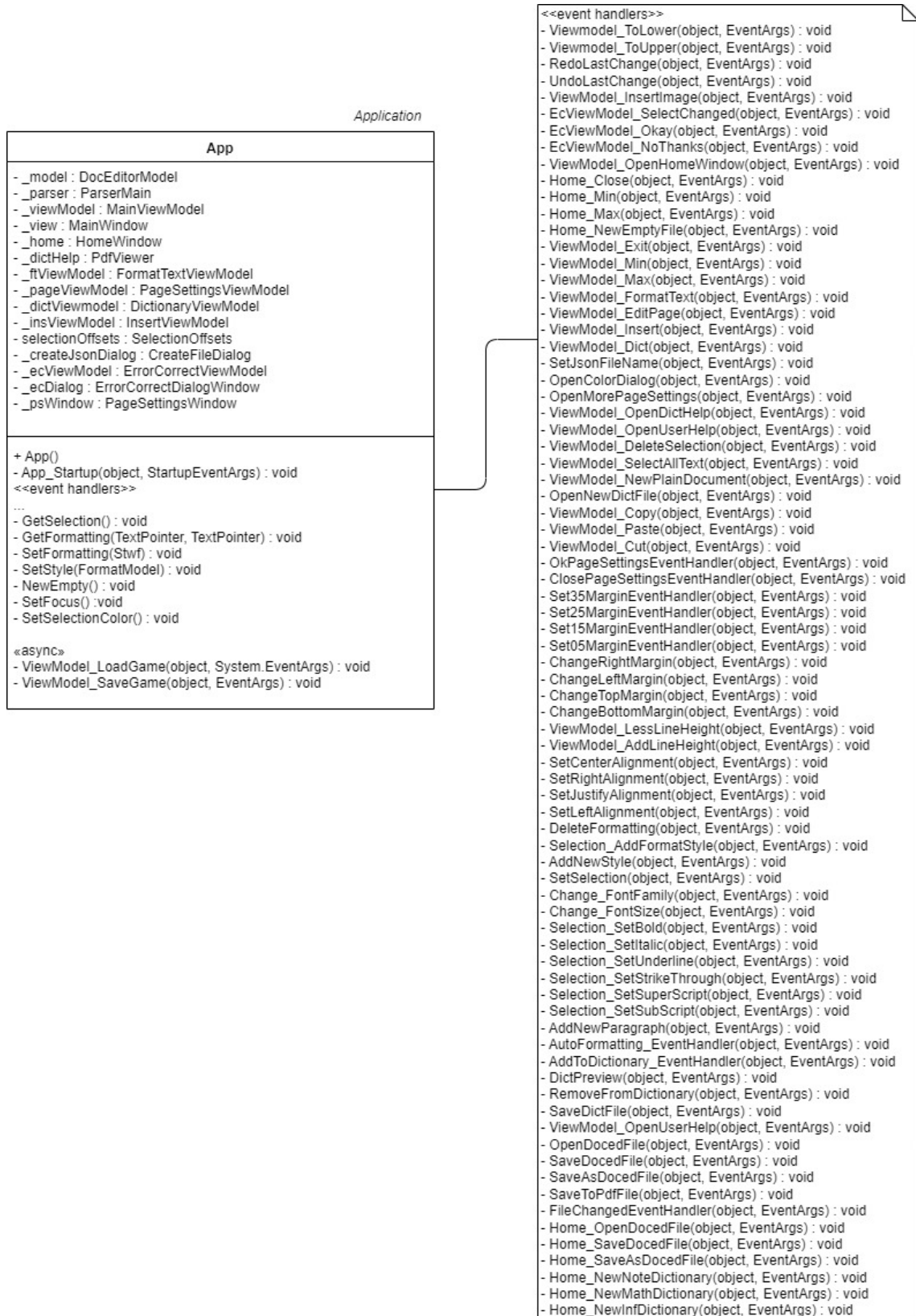
```
1 public int Levenshtein(string str, string dictStr)
2     {
3         //edit distance for str and dictStr
4         int n = str.Length;
5         int m = dictStr.Length;
6         int[,] levTable = new int[n + 1, m + 1];
7         for (int i = 0; i < n + 1; i++)
8         {
9             levTable[i, 0] = i;
10        }
11        for (int i = 1; i < m + 1; i++)
12        {
13            levTable[0, i] = i;
14        }
15
16        for (int i = 1; i < n + 1; i++)
17        {
18            for (int j = 1; j < m + 1; j++)
19            {
20                levTable[i, j] = str[i - 1] == dictStr[j - 1]
21                    ? levTable[i - 1, j - 1]
22                    : Math.Min(levTable[i - 1, j - 1],
23                        Math.Min(levTable[i - 1, j],
24                            levTable[i, j - 1])) + 1;
25            }
26        }
27    }
```

```
25         }
26     }
27     return levTable[str.Length, dictStr.Length];
28 }
```

#### 4.1. forráskód. Levenshtein távolság - C#

Emellett lehetőség van a szótárból elem törlésére (**RemoveElementFromDictionary()**), az adott elemhez tartozó formázás lekérdezésére (**GetFormatting()**), és a szótár tartalmának JSON fájlba való kiírására, JSON fájlból való beolvasására.

### 4.5.7. Környezet



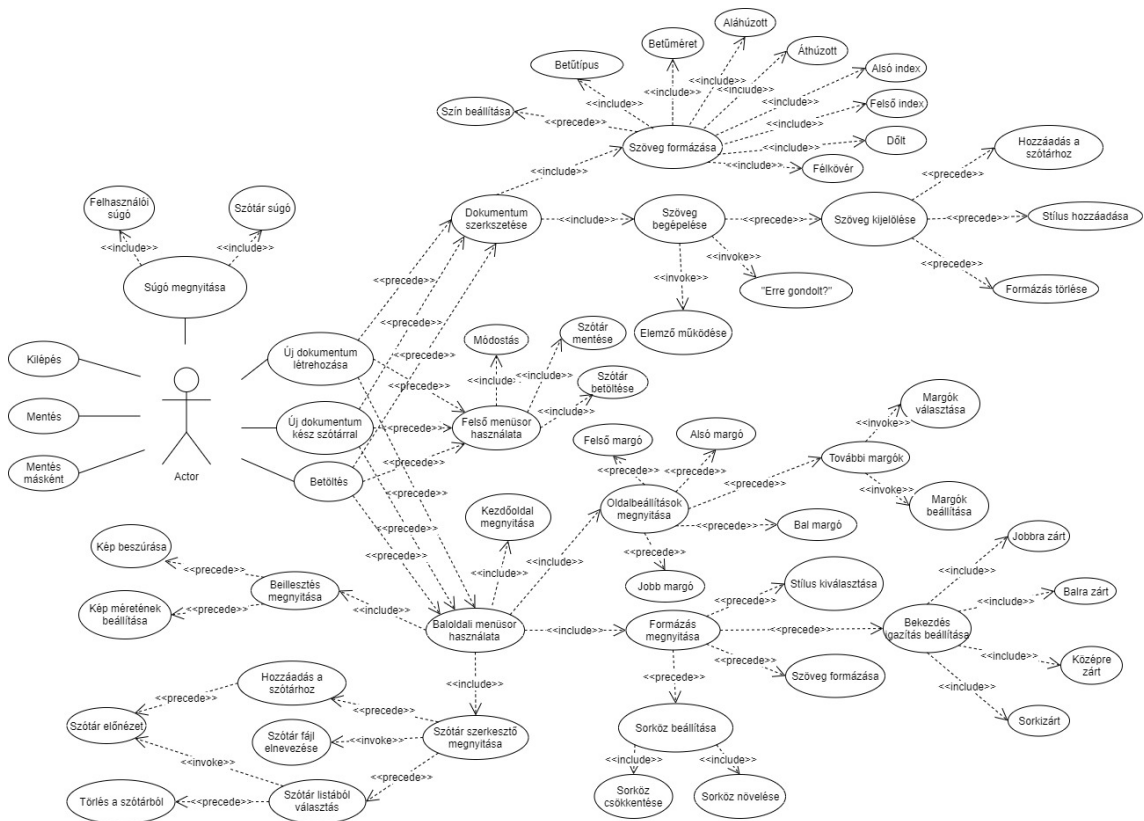
4.15. ábra. A vezérlés osztálydiagramja

Az **App** osztály felel a rétegek példányosításáért és összekötéséért. Az **App\_Startup** hozza létre az alkalmazás rétegeinek példányait, és a rétegek eseményeihez eseményvezérlőket köt.

A nézeten elhelyezett *RichTextBox* vezérlőn bizonyos funkciók hatására történő változásokat ez az osztály teszi lehetővé a modell és az elemző működése alapján. A dialógusablakok megjelenítése, a nézetmodellek közötti váltások, a gombok műveleteinek irányítása is az osztály feladatai közé tartoznak.

## 4.6. Felhasználói esetek

Az alkalmazás egyetlen felhasználóra lett elkészítve, nincsenek különböző felhasználói jogkörök, minden felhasználó egyenrangú. A felhasználó által elvégezhető műveleteket az alábbi ábra mutatja be.



4.16. ábra. A felhasználói esetek

## 4.7. Felhasználói történetek

A program működésének definiálására megfelelő a felhasználói történetek elkészítése, ugyanis az alkalmazás felületein futás közben a felhasználó számára elérhető funkciók és események jól leírhatók ezekkel.

A felhasználói történeteket a következő táblázat szemlélteti:

| Az alkalmazás indítása     |  |
|----------------------------|--|
| AS A                       | felhasználó  |
| I WANT TO                  | elindítani az alkalmazást  |
| GIVEN                      | Az alkalmazás telepítve van.   |
| WHEN                       | Az alkalmazás indítása.  |
| THEN                       | Megjelenik a kezdőoldal.   |
| Az alkalmazás kezdőoldalán |  |
| AS A                       | felhasználó  |
| I WANT TO                  | elkezdeni egy dokumentum szerkesztését   |
| GIVEN                      | Az "Új dokumentum kész szótárral lista tartalmaz elemeket."                                    |
| WHEN                       | Választunk egyet az "Új dokumentum kész szótárral listából".                                   |
| THEN                       | Betöltődik a szótár és egy új üres dokumentum az alkalmazás főoldalán, a kezdőoldal bezáródik. |
| GIVEN                      | A kezdőoldal meg van nyitva.   |
| WHEN                       | Az "Új üres" gombra kattintunk.  |
| THEN                       | Megnyílik a főoldal egy üres dokumentummal, szótár nélkül, a kezdőoldal bezáródik.             |
| GIVEN                      | A kezdőoldal meg van nyitva.   |
| WHEN                       | A "Megnyitás" gombra kattintunk.   |
| THEN                       | Megnyílik a főoldal egy üres dokumentummal, szótár nélkül, a kezdőoldal bezáródik.             |
| GIVEN                      | A kezdőoldal meg van nyitva, még nem mentettünk.   |
| WHEN                       | A "Mentés" gombra kattintunk.  |

|                                |   |
|--------------------------------|---|
| THEN                           | Megnyílik egy dialógusablak, ahol elmenthetjük a dokumentumot.                                  |
| GIVEN                          | A kezdőoldal meg van nyitva, már mentettünk korábban.   |
| WHEN                           | A "Mentés" gombra kattintunk.   |
| THEN                           | A dokumentum a korábban létrehozott fájlba lesz elmentve.                                       |
| GIVEN                          | A kezdőoldal meg van nyitva.  |
| WHEN                           | A "Mentés másként" gombra kattintunk.   |
| THEN                           | Megnyílik egy dialógusablak, ahol elmenthetjük a dokumentumot a megfelelő formátumban.          |
| AS A                           | felhasználó   |
| I WANT TO                      | a kezdőoldal további funkcióit használni  |
| GIVEN                          | A kezdőoldal meg van nyitva.  |
| WHEN                           | A "Súgó" gombra kattintunk.   |
| THEN                           | Megnyílik egy ablak, ahol a programhoz tartozó felhasználói segédlet található pdf formátumban. |
| GIVEN                          | A kezdőoldal meg van nyitva.  |
| WHEN                           | A bezárás gombra kattintunk.  |
| THEN                           | Az alkalmazás leáll.  |
| GIVEN                          | A kezdőoldal meg van nyitva.  |
| WHEN                           | A "tálcára" gombra kattintunk.  |
| THEN                           | Az alkalmazás a tálcára kerül.  |
| <b>Az alkalmazás főoldalán</b> |   |
| AS A                           | felhasználó   |
| I WANT TO                      | használni a bal oldali menüsört   |
| GIVEN                          | A kezdőoldal betöltődött.   |
| WHEN                           | A bal oldali menüsorban a legfelső, házat ábrázoló ikonra kattintunk.                           |
| THEN                           | Megnyílik új ablakban a kezdőoldal.   |
| GIVEN                          | A kezdőoldal betöltődött.   |
| WHEN                           | A bal oldali menüsorban az oldal ikonra kattintunk.   |



|           |   |
|-----------|---|
| THEN      | Megjelenik az oldalak szerkesztéséhez tartozó felület a jobb oldalon.                       |
| GIVEN     | A kezdőoldal betöltődött.   |
| WHEN      | A bal oldali menüsorban a középső ikonra kattintunk.  |
| THEN      | Megjelenik a dokumentum formázásához tartozó felület a jobb oldalon.                        |
| GIVEN     | A kezdőoldal betöltődött.   |
| WHEN      | A bal oldali menüsorban az utolsó előtti, beszúrás ikonra kattintunk.                       |
| THEN      | Megjelenik a beszúráshoz tartozó felület a jobb oldalon.                                    |
| GIVEN     | Nincs még betöltött szótárunk.  |
| WHEN      | A bal oldali menüsorban a legalsó, szótár ikonra kattintunk.                                |
| THEN      | Egy dialógus ablak ugrik fel, amivel létrehozhatunk egy új szótár fájlt a neve megadásával. |
| GIVEN     | Megjelent a szótár elnevezése dialógus ablak.   |
| WHEN      | Megadjuk a nevet és az "OK" gombra kattintunk.  |
| THEN      | A dialógus ablak bezáródik, és megjelenik a szótárhoz tartozó felület a jobb oldalon.       |
| GIVEN     | Van már betöltött szótárunk.  |
| WHEN      | A bal oldali menüsorban a legalsó, szótár ikonra kattintunk.                                |
| THEN      | Megjelenik a szótárhoz tartozó felület a jobb oldalon.                                      |
| AS A      | felhasználó   |
| I WANT TO | használni a felső menüsört  |
| GIVEN     | A "Fájl" menüpont le van nyitva.  |
| WHEN      | A új létrehozására kattintunk.  |
| THEN      | Új üres fájl nyílik meg az aktuális ablakban.   |
| GIVEN     | A "Fájl" menüpont le van nyitva.  |
| WHEN      | A megnyitásra kattintunk.   |
| THEN      | Megnyílik egy új dialógus ablak a betölteni kívánt dokumentum kiválasztásához.              |
| GIVEN     | A "Fájl" menüpont le van nyitva, korábban már elmentettük a dokumentumot.                   |

|       |   |
|-------|---|
| WHEN  | A mentésre kattintunk.  |
| THEN  | A dokumentum mentése megtörténik.   |
| GIVEN | A "Fájl" menüpont le van nyitva, korábban még nem mentettük a dokumentumot. |
| WHEN  | A mentésre kattintunk.  |
| THEN  | Megnyílik egy új dialógus ablak a dokumentum mentéséhez.                    |
| GIVEN | A "Fájl" menüpont le van nyitva.  |
| WHEN  | A mentés másként menüpontra kattintunk.                                     |
| THEN  | Megnyílik egy új dialógus ablak a dokumentum mentéséhez.                    |
| GIVEN | A "Fájl" menüpont le van nyitva.  |
| WHEN  | A kilépés menüpontra kattintunk.  |
| THEN  | Az alkalmazás bezáródik.  |
| GIVEN | A "Szótár" menüpont le van nyitva.  |
| WHEN  | A szótár betöltésére kattintunk.  |
| THEN  | Megnyílik egy új dialógus ablak az új szótár kiválasztásához.               |
| GIVEN | A "Szótár" menüpont le van nyitva.  |
| WHEN  | A szótár mentésére kattintunk.  |
| THEN  | Megnyílik egy új dialógus ablak az aktuális szótár mentéséhez.              |
| GIVEN | A "Szótár" menüpont le van nyitva.  |
| WHEN  | Kiválasztjuk a szótár megnyitása menüpontot.                                |
| THEN  | A szótár kezelő felülete megnyílik a jobb oldali részben.                   |
| GIVEN | A "Módosítás" menüpont le van nyitva.                                       |
| WHEN  | A vissza menüpontra kattintunk.   |
| THEN  | A legutolsó módosítás a dokumentumban visszavonásra kerül.                  |
| GIVEN | A "Módosítás" menüpont le van nyitva.                                       |
| WHEN  | Az újra menüpontra kattintunk.  |

|       |  |
|-------|--|
| THEN  | A legutolsó visszavont művelet újra megtörténik a dokumentumban. |
| GIVEN | A "Módosítás" menüpont le van nyitva.                            |
| WHEN  | A "nagybetűs" menüpontra kattintunk.                             |
| THEN  | A kijelölt szöveg nagybetűsre változik.                          |
| GIVEN | A "Módosítás" menüpont le van nyitva.                            |
| WHEN  | A "kisbetűs" menüpontra kattintunk.                              |
| THEN  | A kijelölt szöveg kisbetűsre változik.                           |
| GIVEN | A "Módosítás" menüpont le van nyitva.                            |
| WHEN  | A kivágás menüpontra kattintunk.                                 |
| THEN  | A dokumentumban kijelölt szöveg kivágásra kerül.                 |
| GIVEN | A "Módosítás" menüpont le van nyitva.                            |
| WHEN  | A másolás menüpontra kattintunk.                                 |
| THEN  | A dokumentumban kijelölt szöveg másolásra kerül.                 |
| GIVEN | A "Módosítás" menüpont le van nyitva.                            |
| WHEN  | A beillesztés menüpontra kattintunk.                             |
| THEN  | A dokumentumba beillesztésre kerül a vágólapra másolt szöveg.    |
| GIVEN | A "Módosítás" menüpont le van nyitva.                            |
| WHEN  | A törlés menüpontra kattintunk.                                  |
| THEN  | A dokumentumban kijelölt szöveg törlődik.                        |
| GIVEN | A "Módosítás" menüpont le van nyitva.                            |
| WHEN  | Az összes kijelölése menüpontra kattintunk.                      |
| THEN  | A dokumentumban kijelölődik az összes szöveg.                    |
| GIVEN | A "Nézet" menüpont le van nyitva.                                |
| WHEN  | Kiválasztjuk a szótár megnyitása menüpontot.                     |
| THEN  | A szótár kezelő felülete megnyílik a jobb oldali részben.        |
| GIVEN | A "Nézet" menüpont le van nyitva.                                |
| WHEN  | Kiválasztjuk a nyomtatás menüpontot.                             |
| THEN  | Megnyílik a nyomtatási beállítások ablak.                        |
| GIVEN | A nyomtatási beállítások ablak meg van nyitva.                   |
| WHEN  | Kiválasztjuk a mentés pdf fájlként menüpontot.                   |

|                                 |   |
|---------------------------------|---|
| THEN                            | A dokumentumot elmentjük pdf fájlként.  |
| GIVEN                           | A "Súgó" menüpont le van nyitva.  |
| WHEN                            | Kiválasztjuk a felhasználói súgót.  |
| THEN                            | Megnyílik új ablakban a a felhasználói segédlet dokumentuma.  |
| GIVEN                           | A "Súgó" menüpont le van nyitva.  |
| WHEN                            | Kiválasztjuk a szótár súgót.  |
| THEN                            | Megnyílik új ablakban a a szótár súgó dokumentuma.  |
| GIVEN                           | A kezdőoldal betöltődött.   |
| WHEN                            | A bezárás gombra kattintunk a jobb felső sarokban.  |
| THEN                            | Az alkalmazás leáll.  |
| GIVEN                           | A kezdőoldal betöltődött.   |
| WHEN                            | A méret növelése gombra kattintunk a jobb felső sarokban.   |
| THEN                            | Az ablak mérete nagyobb lesz.   |
| GIVEN                           | A kezdőoldal betöltődött.   |
| WHEN                            | A tálcára letevő gombra kattintunk a jobb felső sarokban.   |
| THEN                            | Az alkalmazás a tálcára kerül.  |
| <b>A szótár kezelő felülete</b> |   |
| AS A                            | felhasználó   |
| I WANT TO                       | használni a szótár funkcióit  |
| GIVEN                           | Nincs még elem a betöltött szótárban.   |
| WHEN                            | Szó kijelölése után a hozzáadás gombra kattintunk.  |
| THEN                            | Ha van kijelölt szó, akkor az és a hozzá tartozó formázás a szótárba kerül, és megjelenik a listában, ha nem jó a kijelölés, felugó ablakban hibaüzenetet kapunk. |
| GIVEN                           | A betöltött szótár már tartalmaz elemeket.  |
| WHEN                            | Kijelölünk egy elemet a listából.   |
| THEN                            | A törlés gombra kattintással törölhetjük a szótárból a kijelölt elemet.   |
| GIVEN                           | A betöltött szótár már tartalmaz elemeket.  |
| WHEN                            | Kijelölünk egy elemet a listából.   |

|                              |  |
|------------------------------|--|
| THEN                         | A szó és a rá alkalmazott formázás előnézete megjelenik a lista alatti felületen.          |
| GIVEN                        | Nincs kijelölve szó a dokumentumban.   |
| WHEN                         | A hozzáadás gombra kattintunk.   |
| THEN                         | A hibaüzenet felugró ablakban jelenik meg.   |
| GIVEN                        | Nem pontosan egy szót jelöltünk ki a dokumentumban.  |
| WHEN                         | A hozzáadás gombra kattintunk.   |
| THEN                         | A hibaüzenet felugró ablakban jelenik meg.   |
| GIVEN                        | Helyesen kijelöltünk egy szót a dokumentumban.   |
| WHEN                         | A hozzáadás gombra kattintunk.   |
| THEN                         | A szó és a rá alkalmazott formázás hozzáadódik a szótárhoz, az elem megjelenik a listában. |
| <b>Tartalom beillesztése</b> |  |
| AS A                         | felhasználó  |
| I WANT TO                    | képeket beilleszteni a dokumentumba  |
| GIVEN                        | Látható a felület a kép beillesztéséhez.   |
| WHEN                         | Beírjuk a beilleszteni kívánt fájl nevét a szövegdobozba.                                  |
| THEN                         | A kép ikonra kattintva beilleszthetjük a képet a dokumentumba.                             |
| GIVEN                        | Beírtuk a fájlnevet a beszúráshoz.   |
| WHEN                         | A kép ikonra kattintunk.   |
| THEN                         | A képet beilleszthetjük a dokumentumba.  |
| GIVEN                        | Nem írtunk be fájlnevet a beszúráshoz.   |
| WHEN                         | A kép ikonra kattintunk.   |
| THEN                         | Egy dialógus ablak nyílik meg, ahol ki tudjuk választani a képet a beszúráshoz.            |
| <b>A formázás felülete</b>   |  |
| AS A                         | felhasználó  |
| I WANT TO                    | használni formázásokat   |
| GIVEN                        | Van kijelölt szövegrészünk a dokumentumban.  |
| WHEN                         | A felső legördülő listából kiválasztunk egy elemet.  |
| THEN                         | A kiválasztott stílus alkalmazódik a kijelölt szövegre.                                    |

|       |  |
|-------|--|
| GIVEN | Van kijelölt szövegrészünk a dokumentumban.  |
| WHEN  | Az alsó legördülő listából kiválasztunk egy betűtípust.                            |
| THEN  | A kiválasztott betűtípus alkalmazódik a kijelölt szövegre.                         |
| GIVEN | Van kijelölt szövegrészünk a dokumentumban.  |
| WHEN  | Az alsó legördülő listából kiválasztunk egy betűméretet.                           |
| THEN  | A kiválasztott betűméret alkalmazódik a kijelölt szövegre.                         |
| GIVEN | Van kijelölt szövegrészünk a dokumentumban.  |
| WHEN  | A félkövér formázás gombra kattintunk.   |
| THEN  | A kijelölt szöveg félkövér lesz.   |
| GIVEN | Van kijelölt szövegrészünk a dokumentumban.  |
| WHEN  | A dőlt formázás gombra kattintunk.   |
| THEN  | A kijelölt szöveg dőlt lesz.   |
| GIVEN | Van kijelölt szövegrészünk a dokumentumban.  |
| WHEN  | Az aláhúzás gombra kattintunk.   |
| THEN  | A kijelölt szöveg aláhúzott lesz.  |
| GIVEN | Van kijelölt szövegrészünk a dokumentumban.  |
| WHEN  | Az áthúzás gombra kattintunk.  |
| THEN  | A kijelölt szöveg áthúzott lesz.   |
| GIVEN | Van kijelölt szövegrészünk a dokumentumban.  |
| WHEN  | Olyan gombra kattintunk, amihez tartozó formázás már szerepel a kijelölt szövegen. |
| THEN  | A kijelölt szövegről lekerül az adott formázás.                                    |
| GIVEN | Van kijelölt szövegrészünk a dokumentumban.  |
| WHEN  | A szövegszín gombra kattintunk.  |
| THEN  | Megnyílik a színválasztó dialógus ablak, ahol tudunk színt keverni, kiválasztani.  |
| GIVEN | Meg van nyitva a színválasztó dialógus ablak, kiválasztottunk egy színt.           |
| WHEN  | Az OK gombra kattintunk.   |
| THEN  | A kijelölt szöveg a kiválasztott színű lesz, bezáródik a színválasztó ablak.       |
| GIVEN | Meg van nyitva a színválasztó dialógus ablak.                                      |

|       |  |
|-------|--|
| WHEN  | A Mégsem gombra kattintunk.  |
| THEN  | A kijelölt szöveg színe változatlan marad, bezáródik a szín-választó ablak.                                |
| GIVEN | Van kijelölt szövegrészünk a dokumentumban.  |
| WHEN  | Az alsó index gombra kattintunk.   |
| THEN  | A kijelölt szöveg alsó indexbe kerül.  |
| GIVEN | Van kijelölt szövegrészünk a dokumentumban.  |
| WHEN  | A felső index gombra kattintunk.   |
| THEN  | A kijelölt szöveg felső indexbe kerül.   |
| GIVEN | Van kijelölt szövegrészünk a dokumentumban.  |
| WHEN  | A formázás törlése gombra kattintunk.  |
| THEN  | A kijelölt szövegre az alapvető formázási stílus lesz érvényes.  |
| GIVEN | Van kijelölt szövegrészünk a dokumentumban.  |
| WHEN  | A stílus hozzáadása gombra kattintunk.   |
| THEN  | A kijelölt szövegre alkalmazott formázások egy új stílusként bekerülnek a korábbi stílusok közé a listába. |
| GIVEN | A felület meg van nyitva.  |
| WHEN  | A balra zárt igazítás gombra kattintunk.   |
| THEN  | Az aktuális paragrafus igazítása balra zárt lesz.  |
| GIVEN | A felület meg van nyitva.  |
| WHEN  | A jobbra zárt igazítás gombra kattintunk.  |
| THEN  | Az aktuális paragrafus igazítása jobbra zárt lesz.   |
| GIVEN | A felület meg van nyitva.  |
| WHEN  | A középre zárt igazítás gombra kattintunk.   |
| THEN  | Az aktuális paragrafus igazítása középre zárt lesz.  |
| GIVEN | A felület meg van nyitva.  |
| WHEN  | A sorkizárt igazítás gombra kattintunk.  |
| THEN  | Az aktuális paragrafus igazítása sorkizárt lesz.   |
| GIVEN | A felület meg van nyitva.  |
| WHEN  | A sorköz növelése gombra kattintunk.   |
| THEN  | A sorköz nő a dokumentumban.   |

|       |   |
|-------|---|
| GIVEN | A felület meg van nyitva.                                   |
| WHEN  | A sorköz csökkentése gombra kattintunk.                     |
| THEN  | A sorköz csökken a dokumentumban.                           |
| GIVEN | Nincs kijelölt szövegrész a dokumentumban.                  |
| WHEN  | Valamelyik formázáshoz tartozó gombra kattintunk.           |
| THEN  | Az adott formázás alkalmazódik az ezután begépett szövegre. |

#### Az oldal beállításai

|           |  |
|-----------|--|
| AS A      | felhasználó  |
| I WANT TO | oldalmargót beállítani a dokumentumhoz   |
| GIVEN     | A felület betöltődött.   |
| WHEN      | A szövegdobozokba írjuk a beállítani kívánt értéket.   |
| THEN      | A szövegdoboz melletti gombra kattintva beállítjuk az adott (felső, alsó, bal, jobb) margót a dokumentumhoz. |
| GIVEN     | A felület betöltődött.   |
| WHEN      | A "... " ikonra kattintunk.  |
| THEN      | Megnyílik a dialógus ablak az oldalmargó további beállításaihoz.   |

#### Erre gondolt? dialógus ablak

|           |   |
|-----------|---|
| AS A      | felhasználó   |
| I WANT TO | használni az "Erre gondolt?" funkciót   |
| GIVEN     | Megnyílt a dialógus ablak.  |
| WHEN      | A listából kiválasztunk egy szót.   |
| THEN      | A lista alatti felületen megjelenik a kiválasztott szó formázott előnézete.   |
| GIVEN     | Kiválasztottunk egy szót a listából.  |
| WHEN      | Az OK gombra kattintunk.  |
| THEN      | A dialógus ablak bezáródik, és a dokumentumunkban az utoljára begépett szó kijavítódik az általunk kiválasztott szóra, és az ahhoz tartozó formázás lesz rá alkalmazva a szótárból. |
| GIVEN     | Megnyílt a dialógus ablak.  |



|  |   |
|--|---|
| WHEN                                   | A MÉGSEM gombra kattintunk.   |
| THEN                                   | A dialógus ablak bezáródik, nem történik változás a dokumentumban.                      |
| <b>Oldalbeállítások dialógus ablak</b> |   |
| AS A                                   | felhasználó   |
| I WANT TO                              | használni az "Erre gondolt?" funkciót   |
| GIVEN                                  | Megnyílt a dialógus ablak.  |
| WHEN                                   | Az alsó gomb listából az egyikre kattintunk.  |
| THEN                                   | A dialógus ablak bezáródik, és a gombon feltüntetett margó állítódik be a dokumentumra. |
| GIVEN                                  | Megnyílt a dialógus ablak.  |
| WHEN                                   | A fenti szövegdozok melletti nyilakra kattintunk.                                       |
| THEN                                   | A szövegdozokban nő vagy csökken az aktuális margó értéke 0,1-gyel.                     |
| GIVEN                                  | Beállítottuk a margókat.  |
| WHEN                                   | Az OK gombra kattintunk.  |
| THEN                                   | A dialógus ablak bezáródik, és a dokumentumra alkalmazódik a beállított oldalmargó.     |
| GIVEN                                  | Megnyílt a dialógus ablak.  |
| WHEN                                   | A MÉGSEM gombra kattintunk.   |
| THEN                                   | A dialógus ablak bezáródik, nem történik változás a dokumentumban.                      |
| <b>Új szótár létrehozása</b>           |   |
| AS A                                   | felhasználó   |
| I WANT TO                              | létrehozni egy új fájlt a szótárnak   |
| GIVEN                                  | Megnyílt a dialógus ablak.  |
| WHEN                                   | Beírjuk a fájlnevet a szövegdozba.  |
| THEN                                   | Az Ok gombra kattintva létrehozhatjuk a fájlt.  |
| GIVEN                                  | A szövegdozba már beírtuk a fájlnevet.  |
| WHEN                                   | Az OK gombra kattintunk.  |
| THEN                                   | A dialógus ablak bezáródik, és betöltődik a létrehozott fájl a szótárhoz.               |

|                                |  |
|--------------------------------|--|
| GIVEN                          | Nem adtunk meg fájlnévet, vagy nem helyes fájlnévet adtunk meg.                |
| WHEN                           | Az OK gombra kattintunk.   |
| THEN                           | Egy felugó ablak jelzi a hibát.  |
| <b>Dokumentum szerkesztése</b> |  |
| AS A                           | felhasználó  |
| I WANT TO                      | dokumentumot szerkeszteni  |
| GIVEN                          | A dokumentumhoz tartozó papír a felületen van.                                 |
| WHEN                           | Gépelünk a billentyűzeten.   |
| THEN                           | A begépelte szöveg megjelenik a dokumentumon.                                  |
| GIVEN                          | Az utoljára begépelte szó szerepel a szótárban.                                |
| WHEN                           | Leütünk egy szóközt.   |
| THEN                           | A begépelte szóra alkalmazódik a formázás a szótárból.                         |
| GIVEN                          | Az utoljára begépelte szó nem a szótárban, de a hibajavító talál egyezést.     |
| WHEN                           | Leütünk egy szóközt.   |
| THEN                           | Felugrik az "Erre gondolt?" dialógus ablak.                                    |
| GIVEN                          | Az utoljára begépelte szó nem a szótárban, és a hibajavító sem talál egyezést. |
| WHEN                           | Leütünk egy szóközt.   |
| THEN                           | Folytathatjuk a gépelést vagy a begépelte szavak szerkesztését.                |
| GIVEN                          | A dokumentumhoz tartozó papír a felületen van.                                 |
| WHEN                           | Leültünk egy entert.   |
| THEN                           | Új paragrafust hozunk létre a dokumentumunkban.                                |
| GIVEN                          | A dokumentumhoz tartozó papír a felületen van.                                 |
| WHEN                           | A <i>ctrl + z</i> billentyűkombinációt ütjük le.                               |
| THEN                           | A legutolsó módosítás visszavonásra kerül a dokumentumban.                     |
| GIVEN                          | A dokumentumhoz tartozó papír a felületen van.                                 |
| WHEN                           | A <i>ctrl + y</i> billentyűkombinációt ütjük le.                               |

|       |  |
|-------|--|
| THEN  | A legutolsó visszavont módosítás újra megtörténik a dokumentumban. |
| GIVEN | A dokumentumban van kijelölve szöveg.                              |
| WHEN  | A <i>ctrl + c</i> billentyűkombinációt ütjük le.                   |
| THEN  | A kijelölt szöveg a vágólapra másolódik.                           |
| GIVEN | A dokumentumhoz tartozó papír a felületen van.                     |
| WHEN  | A <i>ctrl + v</i> billentyűkombinációt ütjük le.                   |
| THEN  | A vágólapról beillesztésre kerül a szöveg a dokumentumba.          |
| GIVEN | A dokumentumban van kijelölve szöveg.                              |
| WHEN  | A <i>ctrl + x</i> billentyűkombinációt ütjük le.                   |
| THEN  | Megtörténik a kijelölt szöveg kivágása.                            |
| GIVEN | A dokumentumhoz tartozó papír a felületen van.                     |
| WHEN  | A <i>ctrl + a</i> billentyűkombinációt ütjük le.                   |
| THEN  | Megtörténik a teljes szöveg kijelölése.                            |
| GIVEN | A dokumentumban van kijelölve szöveg.                              |
| WHEN  | A <i>ctrl + b</i> billentyűkombinációt ütjük le.                   |
| THEN  | A kijelölt szöveg félkövérre változik.                             |
| GIVEN | A dokumentumban van kijelölve szöveg.                              |
| WHEN  | A <i>ctrl + i</i> billentyűkombinációt ütjük le.                   |
| THEN  | A kijelölt szöveg dőltre változik.                                 |
| GIVEN | A dokumentumban van kijelölve szöveg.                              |
| WHEN  | A <i>ctrl + u</i> billentyűkombinációt ütjük le.                   |
| THEN  | A kijelölt szöveg aláhúzotttra változik.                           |
| GIVEN | A dokumentumhoz tartozó papír a felületen van.                     |
| WHEN  | A <i>ctrl + e</i> billentyűkombinációt ütjük le.                   |
| THEN  | A szöveg középre igazítódik.                                       |
| GIVEN | A dokumentumhoz tartozó papír a felületen van.                     |
| WHEN  | A <i>ctrl + j</i> billentyűkombinációt ütjük le.                   |
| THEN  | A szöveg sorkizártra igazítódik.                                   |
| GIVEN | A dokumentumhoz tartozó papír a felületen van.                     |
| WHEN  | A <i>ctrl + r</i> billentyűkombinációt ütjük le.                   |
| THEN  | A szöveg jobbra igazítódik.  |

|       |  |
|-------|--|
| GIVEN | A dokumentumhoz tartozó papír a felületen van.   |
| WHEN  | A <i>ctrl + l</i> billentyűkombinációt ütjük le. |
| THEN  | A szöveg balra igazítódik.                       |

4.1. táblázat. Felhasználói történetek

## 4.8. Tesztelés

Az alkalmazás tesztelését több részben végeztem el, mivel a program nagy méretű, és funkcionalitását tekintve jól elhatárolódnak az azt felépítő modulok, projektek. A tesztek során a program logikai részének működését ellenőriztem. Alapos tesztelésnek vetettem alá az elemző és a modell fontos algoritmusait, adatkonverziós műveleteit, különböző számításokat és átalakításokat. A teszteléshez Unit teszteket használtam.

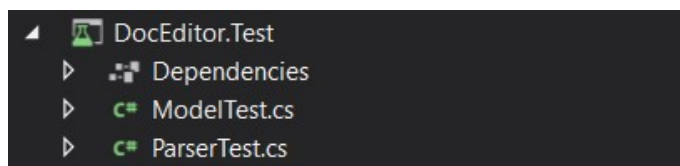
A tesztelést közvetlenül az adott részek elkészülte után végeztem, így az esetlegesen felmerülő hibákat a tesztek segítségével gyorsan tudtam javítani, és meg tudtam győződni róla, hogy az implementált műveletek minden lehetséges esetben jól működnek.

A grafikus felület tesztelését is a fejlesztéssel párhuzamosan végeztem, illetve a program elkészülését követő tesztidőszakban folytattam. A tesztelés közben igyekeztem előidézni az összes lehetséges szélsőséges felhasználói műveletet, ami következében az alkalmazás hibát adhat, az összes megtalált nem kezelt ilyen esetet ki is tudtam sikeresen javítani.

### 4.8.1. Tesztkörnyezet

A tesztelés az alábbi környezetekben történt:

- Unit tesztek: Microsoft Visual Studio
- Grafikus felület tesztje: Asus VivoBook S5, Intel Core i5 négymagos, 1,8 Ghz processzor, 8 GB RAM, Windows 11 operációs rendszer



4.17. ábra. A tesztprojekt

A teszteléshez a *DocEditor.Test* projektet készítettem el. A tesztek futtatásához először le kell fordítani a projektet, majd a Test Explorerben lehet megnézni a futtatás után a tesztesetek eredményeit.

#### 4.8.2. Manuális tesztelés

A felhasználói felület és a felhasználói felületről elérhető funkciók, műveletek tesztelését manuális tesztekkel végeztem, minden elkészült új funkció után közvetlenül és a projekt elkészülése utáni tesztelési időszakban.

Szemponatok, amiket használtam a megjelenítés manuális tesztelése folyamán:

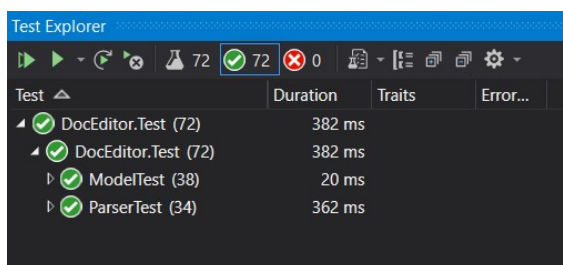
- A megjelenített tartalom teljes egészében látsszon minden vezérlőben.
- Az ablak méretezése biztosítsa minden esetben minden felületi elem jó elhelyezkedését, láthatóságát, ezeket használni is lehessen funkciójuknak megfelelően.
- A felület elemei megfelelően reagáljanak a felhasználói interakciókra, szélsőséges használati esetek mellett se adjon hibát, a felhasználó ne tudja akadályozni a program futását. Azok a felhasználói interakciók, amik hibát tudnának okozni, legyenek ellenőrzéssel biztosítva.
- Az alkalmazás megjelenítésének stílusa legyen egységes, letisztult, könnyen átlátható.
- Beviteli mezők esetén a felhasználó csak megfelelő értékeket tudjon megadni, ezeknek ellenőrzése real-time történjen.

A manuális tesztelés során minden funkció tesztelésekor igyekeztem olyan eseteket kiváltani, amik lehetségesen hibát eredményeznek. Számos funkció hibáit felismertem hamar ennek köszönhetően, és a javításukat is el tudtam végezni.

### 4.8.3. Unit tesztek

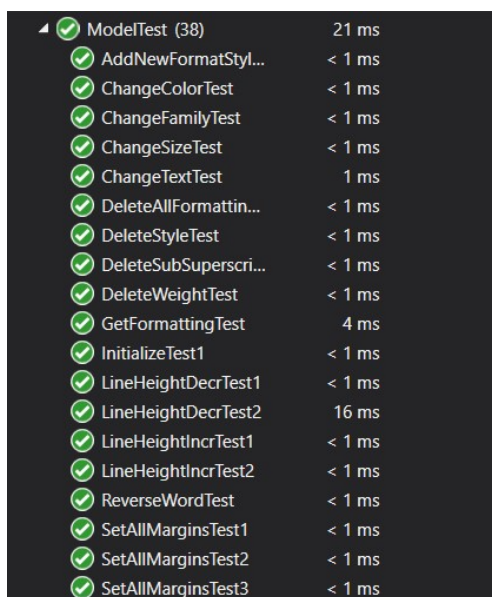
A teszt projekt MSTest keretrendszert használ, amit a Microsoft fejlesztett, és talán az egyik legelterjedtebb teszt keretrendszer. A keretrendszer attribútumokat biztosít a teszt metódusok és teszt osztályok megjelölésére. A tesztek eredményeinek ellenőrzésére az Assert osztály különböző függvényeit használhatjuk. A teszteléshez felhasznált metódusok:

- Assert.AreEqual()
- Assert.IsNull()
- Assert.IsNotNull()

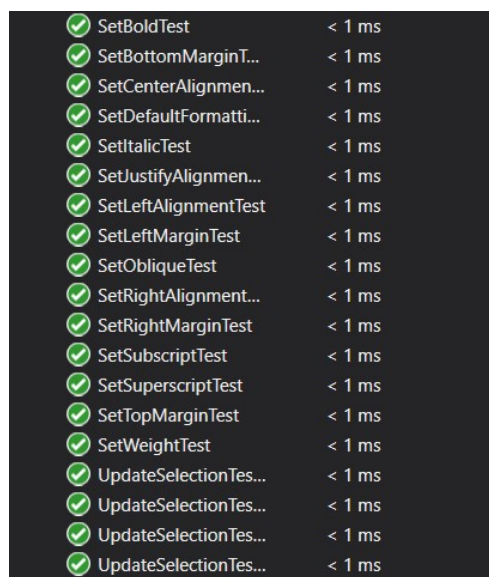


4.18. ábra. A tesztek futása

A modell tesztelésénél a modell összes osztályát leteszteltem, végigjárva minden fontos metódusukat. Mivel a modell csak alapvető adatkonverziókat és egyszerűbb műveleteket végez el, a tesztelés során nagyon kevés hibával találkoztam, azokat is egyből ki tudtam javítani. A tesztesetek a következő ábrákon láthatók:

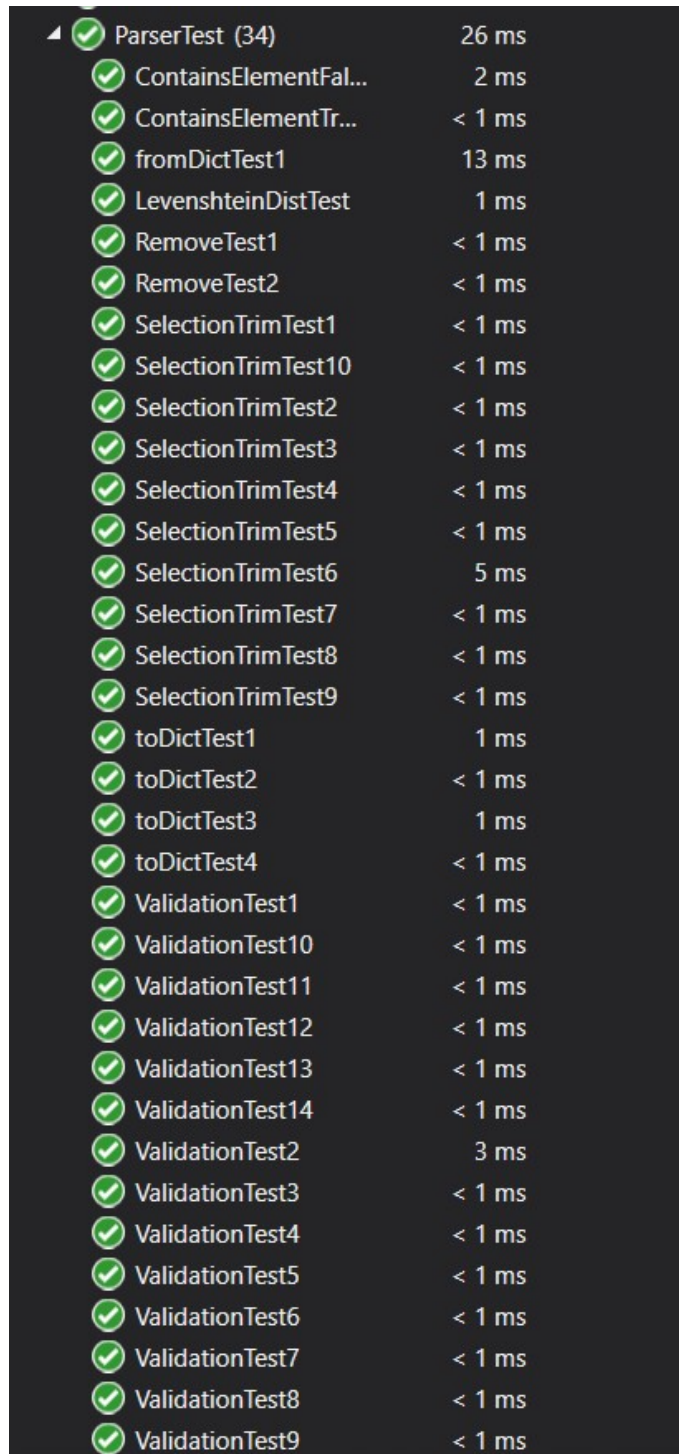


4.19. ábra. Modell tesztek



4.20. ábra. Modell tesztek

Az elemző tesztelése érdekesebb feladatnak bizonyult. Itt jól kellett modellezni különböző eseteket, amikkel az elemző találkozhat, hogy a reakciót körültekintően tudjam tesztelni minden esetre. Az elemző tartalmaz több bonyolultabb algoritmust is, ezeknek a tesztelése különösen fontos volt. Az elemző tesztesei a következő ábrákon láthatók:



|                         |        |
|-------------------------|--------|
| ✓ ParserTest (34)       | 26 ms  |
| ✓ ContainsElementFal... | 2 ms   |
| ✓ ContainsElementTr...  | < 1 ms |
| ✓ fromDictTest1         | 13 ms  |
| ✓ LevenshteinDistTest   | 1 ms   |
| ✓ RemoveTest1           | < 1 ms |
| ✓ RemoveTest2           | < 1 ms |
| ✓ SelectionTrimTest1    | < 1 ms |
| ✓ SelectionTrimTest10   | < 1 ms |
| ✓ SelectionTrimTest2    | < 1 ms |
| ✓ SelectionTrimTest3    | < 1 ms |
| ✓ SelectionTrimTest4    | < 1 ms |
| ✓ SelectionTrimTest5    | < 1 ms |
| ✓ SelectionTrimTest6    | 5 ms   |
| ✓ SelectionTrimTest7    | < 1 ms |
| ✓ SelectionTrimTest8    | < 1 ms |
| ✓ SelectionTrimTest9    | < 1 ms |
| ✓ toDictTest1           | 1 ms   |
| ✓ toDictTest2           | < 1 ms |
| ✓ toDictTest3           | 1 ms   |
| ✓ toDictTest4           | < 1 ms |
| ✓ ValidationTest1       | < 1 ms |
| ✓ ValidationTest10      | < 1 ms |
| ✓ ValidationTest11      | < 1 ms |
| ✓ ValidationTest12      | < 1 ms |
| ✓ ValidationTest13      | < 1 ms |
| ✓ ValidationTest14      | < 1 ms |
| ✓ ValidationTest2       | 3 ms   |
| ✓ ValidationTest3       | < 1 ms |
| ✓ ValidationTest4       | < 1 ms |
| ✓ ValidationTest5       | < 1 ms |
| ✓ ValidationTest6       | < 1 ms |
| ✓ ValidationTest7       | < 1 ms |
| ✓ ValidationTest8       | < 1 ms |
| ✓ ValidationTest9       | < 1 ms |

4.21. ábra. Az elemző tesztjei

A modell és az elemző tesztelésénél is jól elkülönítve teszteltem a különböző műveleteket, algoritmusokat, a műveletek alkalmazásának lehetséges változatait. Így a tesztprojektem jól átlátható, sok tesztesetet tartalmaznak, viszont könnyen értelmezhetőek és kezelhetőek ezáltal a tesztek.

## 4.9. Verziókezelés

Mivel az implementáció több lépésben történt, szükséges volt, hogy a program egyes állapotai jól nyomonkövethetők legyenek. A projekt verziókezeléséhez **Git** elosztott verziókezelő rendszert használtam, amit integráltan támogat a **GitHub** projektvezető szolgáltatás.

A Git verziókövető rendszer az összes korábbi változatot eltárolja és lehetőséget ad ezeknek a letöltésére. Emellett a fejlesztés futhat több ágon az úgynevezett *branchek* használatával. Ezeken a fejlesztési ágakon nagyobb volumenű feladatokat, funkciókat implementálhatunk elkülönítve a fő fejlesztési ágtól, majd az implementáció elkészültekor egyesíthetjük őket egymással. Így az egyes funkciók módosításainak követése, esetleg a módosítások elvetése jelentősen könnyebbé válik. A Git lehetőséget nyújt a változtatások egyszerű visszavonására, korábbi változatokra való visszatérésre. Ezek nagy előnyt jelentettek a fejlesztés során.



## 5. fejezet

# Összegzés

A szakdolgozatom keretein belül egy jól használható, sok funkcióval rendelkező alkalmazást hoztam létre, ami számos további lehetőséget hordoz magában.

Külön öröm számomra, hogy egy olyan ötletet valósíthattam meg, ami saját tapasztalatok alapján született meg. Igaz, hogy ez nem a végleges verziója a programnak, de egy olyan alkalmazás fejlesztését kezdhettem el, ami korábban sokáig sokszor csak "milyen jó lenne, ha használhatnék ilyen programot" gondolatként élt a fejemben.

### 5.1. Továbbfejlesztési lehetőségek

Matematikai témájú dokumentumokhoz nagyon előnyös funkció lenne, ha az alkalmazáson belül támogatott lenne a komolyabb egyenletek beszúrása és szerkesztése. Ennek az implementálása nem kis feladat, és nagy kihívást is jelent. Jelenleg a képek beszúrása jól helyettesíti ezt a funkciót, de kényelmesebb dokumentumszerkesztés érdekében a közeljövőben tervezett bővítések között tartom számon ezt a továbbfejlesztési lehetőséget.

A képek beszúrásán kívül táblázatok, különböző alakzatok, ábrák beszúrása és szerkesztése is támogatott a FlowDocumentum-ban, így ezek implementálása is tervben van.

Felsorolások és listák készítése, különböző listaszintekkel.

A dolgozat keretein belül csak rtf formátumban készítettem el a mentést, ezt könnyen lehet pdf fájlá konvertálni, például online toolok segítségével, azonban az alkalmazáson belüli pdf fájlba történő mentés funkció is hasznos lenne.

Talán a legnehezebben megvalósítható továbbfejlesztés a dokumentum többoldalasítása. A FlowDocument támogatja ezt a lehetőséget, viszont a WPF-ben nincsen olyan vezérlő, amivel több oldalon megjelenítve tudnánk szerkeszteni a FlowDocumentet. A probléma megoldásán szinte a fejlesztés teljes ideje alatt gondolkoztam, azonban folyamatos akadályokba ütköztem. A leginkább megvalósíthatónak tűnő megoldás az volt, hogy új sor hozzáadása esetén a sorokon végigiterálva összeadom azoknak a magasságát (ami a legnagyobb betűméretből és a sorközből adódik), és ez alapján döntök arról, hogy új oldalt kell-e létrehozni. A vezérlő ebben az esetben is lehetett volna egy RichTextBox, amin mindig az éppen aktuális oldal tartalmát jeleníthettem volna meg. Ez a megoldás amiatt nem lehetséges, hogy a FlowDocument-ből az ehhez szükséges magasság értékek nem kérdezhetők le a sorokból. A problémára jó megoldási ötletnek tűnik egy saját vezérlő elkészítése, ami támogatja a FlowDocument többoldalas nézetét és szerkesztését is.

Az eddig említett lehetőségeken túl az elemzőt is lehet fejleszteni új funkciókkal, vagy az "Erre gondolt?" funkció algoritmusának finomításával.

## A. függelék

# Súgó a szótár használatához

### Általános

A szótár a korábban formázottan elmentett szavak jegyzéke. A szótárba felvett elemeket a menüsorból a szótár kiválasztása után a jobb oldalon látható listában lehet elérni. A listában a szótárban szereplő szavak láthatók, és az elemek kiválasztásakor az alatta található felületen jelenik meg a formázott szó előnézete.

A szótár tartalma megtekinthető még a megfelelő Json fájl betöltésével is, ami strukturált formában tárolja el a szótárba felvett szavakat, az azokra vonatkozó formázásokat és az egyes elemek gyakoriságát, ami azt jelenti, hogy hányszor alkalmazták őket korábban a dokumentumban. A szótárhoz tartozó fájl a program használata közben folyamatosan frissül, mindig a legújabb változat látható benne a szótárról.

### Jelölések

#### Formázás

- dőlt - "I"
- félkövér - "B"
- aláhúzott - "U"
- alsó index - "Q"

- felső index - "P"

### Json fájl olvasása

- "Str" - a szó maga
- "Formatting" - minden karakterre külön vonatkozó formázási információ
  - Egy sor felépítése: "betűstílus (dőlt, ferde, normál), betűtípus (félkövér), betűtípus család, alsó index/felső index, betűméret, szín"
  - Az első sorban a leggyakoribb formázások szerepelnek, a következő sorokban pedig a szó karaktereire vonatkozók. Ahol "\_" szerepel a sorban, oda a leggyakoribb formázási információ érvényes.
- "Frequency" - Az adott elem használatának gyakorisága

## Módosítás

A szótár elemeinek törléséhez válassza ki az eltávolítani kívánt szót, majd kattintson a "TÖRLÉS" gombra.

A felugró ablakban a művelet jóváhagyása után a kiválasztott elem eltávolításra kerül a szótárból.

Új elemek felvételéhez jelölje ki a dokumentumban a készre formázott szót teljes egészében, majd kattintson a "HOZZÁADÁS" gombra.

Az új elem a szótárba kerül, és a listában is látható lesz.

# Irodalomjegyzék

- [1] Cserép Máté. „Eseményvezérelt alkalmazások előadásjegyzet”. ().
- [2] Cserép Máté. *Objektumorientált nyelvek és keretrendszerek*. URL: [https://mcserep.web.elte.hu/data/education/2012-2013-2\\_NET1\\_Gyakorlat\\_01.pdf](https://mcserep.web.elte.hu/data/education/2012-2013-2_NET1_Gyakorlat_01.pdf).
- [3] Baeldung. *Levenshtein distance*. 2021. URL: <https://www.baeldung.com/cs/levenshtein-distance-computation> (elérés dátuma 2022. 05. 20.).
- [4] Sol Kim. „An Efficient Autocorrect Algorithm Using Genetic Algorithm”. (). URL: <https://www.actonscholars.org/wp-content/uploads/2020/10/An-Efficient-Autocorrect-Algorithm-Using-Genetic-Algorithm.pdf> (elérés dátuma 2022. 05. 18.).
- [5] Wikipedia. *.NET Framework version history*. 2021. URL: [https://en.wikipedia.org/wiki/.NET\\_Framework\\_version\\_history](https://en.wikipedia.org/wiki/.NET_Framework_version_history) (elérés dátuma 2022. 05. 20.).
- [6] WPF Tutorial. *The OpenFileDialog*. URL: <https://www.wpf-tutorial.com/dialogs/the-openfiledialog/> (elérés dátuma 2022. 05. 10.).
- [7] WPF Tutorial. *The MessageBox*. URL: <https://www.wpf-tutorial.com/dialogs/the-messagebox/> (elérés dátuma 2022. 05. 10.).
- [8] Microsoft. *Reference Source*. 2022. URL: <https://referencesource.microsoft.com/> (elérés dátuma 2022. 05. 20.).
- [9] Microsoft. *RichTextBox Class*. 2022. URL: <https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.richtextbox?view=windowsdesktop-6.0> (elérés dátuma 2022. 04. 22.).
- [10] Microsoft. *Flow Document Overview*. 2020. URL: <https://docs.microsoft.com/en-us/dotnet/desktop/wpf/advanced/flow-document-overview?view=netframeworkdesktop-4.8> (elérés dátuma 2022. 05. 23.).

- [11] Microsoft. *RichTextBox Overview*. 2022. URL: <https://docs.microsoft.com/en-us/dotnet/desktop/wpf/controls/richtextbox-overview?view=netframeworkdesktop-4.8> (elérés dátuma 2022. 05. 24.).
- [12] Microsoft. *How to serialize and deserialize (marshal and unmarshal) JSON in .NET*. 2022. URL: <https://docs.microsoft.com/en-us/dotnet/standard/serialization/system-text-json-how-to?pivots=dotnet-6-0> (elérés dátuma 2022. 05. 12.).
- [13] Microsoft. *Dialog boxes overview (WPF .NET)*. 2022. URL: <https://docs.microsoft.com/en-us/dotnet/desktop/wpf/windows/dialog-boxes-overview?view=netdesktop-6.0> (elérés dátuma 2022. 05. 06.).
- [14] Microsoft. *.NET documentation*. URL: <https://docs.microsoft.com/en-us/dotnet/> (elérés dátuma 2022. 03. 04.).
- [15] Microsoft. *Csharp documentation*. URL: <https://docs.microsoft.com/en-us/dotnet/csharp/> (elérés dátuma 2022. 03. 04.).
- [16] Microsoft. *How to: Save, Load, and Print RichTextBox Content*. 2022. URL: <https://docs.microsoft.com/en-us/dotnet/desktop/wpf/controls/how-to-save-load-and-print-richtextbox-content?view=netframeworkdesktop-4.8> (elérés dátuma 2022. 05. 22.).
- [17] Microsoft. *Documents in WPF*. 2022. URL: <https://docs.microsoft.com/en-us/dotnet/desktop/wpf/advanced/documents-in-wpf?view=netframeworkdesktop-4.8> (elérés dátuma 2022. 03. 18.).
- [18] Cserép Máté. *2. Beadandó dokumentáció*. URL: [https://mcserep.web.elte.hu/data/education/2021-2022-1\\_EVA/elte\\_eva\\_feladatok\\_2\\_minta.pdf](https://mcserep.web.elte.hu/data/education/2021-2022-1_EVA/elte_eva_feladatok_2_minta.pdf).
- [19] *Eseményvezérelt alkalmazások fejlesztése*. URL: <https://people.inf.elte.hu/gt/eva/eva.html>.
- [20] ELTE IK. „Szoftvertechnológia előadásjegyzet”. ().
- [21] *WPF Tutorial*. URL: <https://www.wpf-tutorial.com/about-wpf/what-is-wpf/>.

# Ábrák jegyzéke

|   |    |
|---|----|
| 2.1. Rekurzív formula a Levenshtein távolság kiszámításához . . . . . | 9  |
| 3.1. Az alkalmazás kezdőoldala . . . . .                              | 14 |
| 3.2. Amit a főoldalon látunk a megnyitása után . . . . .              | 15 |
| 3.3. Az alkalmazás menüje . . . . .                                   | 16 |
| 3.4. Fájl menüpont . . . . .  | 17 |
| 3.5. Szótár menüpont . . . . .  | 17 |
| 3.6. Módosítás menüpont . . . . .                                     | 17 |
| 3.7. Nézet menüpont . . . . .   | 18 |
| 3.8. Súgó menüpont . . . . .  | 18 |
| 3.9. Oldalbeállítások . . . . .                                       | 19 |
| 3.10. További lehetőségek . . . . .                                   | 20 |
| 3.11. Szövegformázás felület . . . . .                                | 21 |
| 3.12. Ablak szín kiválasztásához . . . . .                            | 22 |
| 3.13. Szövegszerkesztési lehetőségek . . . . .                        | 22 |
| 3.14. Képek beszúrása . . . . .                                       | 23 |
| 3.15. Dólt . . . . .  | 24 |
| 3.16. Felső index . . . . .   | 24 |
| 3.17. A felugró ablak . . . . .                                       | 25 |
| 4.1. Az alkalmazás csomagdiagramja . . . . .                          | 32 |
| 4.2. A forrásprojektek szerkezete . . . . .                           | 34 |
| 4.3. A nézethez tartozó projekt . . . . .                             | 35 |
| 4.4. A Persistence csomag osztálydiagramja . . . . .                  | 36 |
| 4.5. A Model csomag . . . . .   | 37 |
| 4.6. A formázásért felelős osztályok . . . . .                        | 39 |
| 4.7. A kijelölésért felelős osztályok . . . . .                       | 40 |
| 4.8. A fő nézetmodell osztálydiagramja . . . . .                      | 42 |

|   |    |
|---|----|
| 4.9. A szövegek szerkesztéséhez tartozó nézetmodell osztálydiagramja . . .  | 43 |
| 4.10. A képek beszúrásához tartozó nézetmodell osztálydiagramja . . . . .   | 44 |
| 4.11. Az oldalbeállításokhoz tartozó nézetmodell osztálydiagramja . . . . . | 45 |
| 4.12. A szótárhoz tartozó nézetmodell osztálydiagramja . . . . .            | 46 |
| 4.13. Az "Erre gondolt?" funkcióhoz tartozó nézetmodell osztálydiagramja    | 47 |
| 4.14. Az elemző osztálydiagramja . . . . .                                  | 48 |
| 4.15. A vezérlés osztálydiagramja . . . . .                                 | 52 |
| 4.16. A felhasználói esetek . . . . .                                       | 53 |
| 4.17. A tesztprojekt . . . . .  | 68 |
| 4.18. A tesztek futása . . . . .  | 69 |
| 4.19. Modell tesztek . . . . .  | 69 |
| 4.20. Modell tesztek . . . . .  | 69 |
| 4.21. Az elemző tesztjei . . . . .  | 70 |



# Táblázatok jegyzéke

|   |    |
|---|----|
| 2.1. Levenshtein távolság táblázattal - első lépés . . . . .          | 9  |
| 2.2. Levenshtein távolság táblázattal . . . . .                       | 10 |
| 3.1. Az alkalmazás felületein található gombok és funkcióik . . . . . | 28 |
| 4.1. Felhasználói történetek . . . . .                                | 67 |

# Algoritmusjegyzék

|    |  |    |
|----|--|----|
| 1. | Rekurzív algoritmus Levenshtein távolság kiszámítására . . . . . | 10 |
| 2. | Iteratív algoritmus mátrixszal . . . . .                         | 11 |

# Forráskódjegyzék

|  |    |
|--|----|
| 4.1. Levenshtein távolság - C# . . . . . | 50 |
|--|----|