

## Házi Feladat

**Beadási határidő:  
2025.12.19 23:59:59**

# Szöveges fájlok összehasonlítása Jaccard-hasonlósággal

## Cél

A program célja, hogy két szöveges fájlt (TXT) hasonlítsa össze, és meghatározza, hogy hány százalékban egyeznek. A rendszer objektumorientált programozás (OOP) elveket követ, és tartalmaz kompozíciót, valamint absztrakt osztályokat.

## Főbb funkciók

### 1. Szöveges fájlok betöltése és előkészítése

A program képes két TXT fájl betöltésére és előkészítésére az összehasonlításhoz. A fájlok beolvasása után a szövegeket megfelelően elő kell készíteni az összehasonlításhoz, például a szóközök eltávolításával és a kis- és nagybetűk közötti eltérések figyelmen kívül hagyásával.

### 2. Szöveg összehasonlítása

A rendszer összehasonlítja a fájlok tartalmát, és kiszámítja, hogy hány százalékban egyeznek a szövegek. Az összehasonlítás szó szintjén történik, és a Jaccard-hasonlóság elvét alkalmazza.

### 3. Kompozíció

A programban kompozíciót alkalmazunk. A FileComparer osztály egy TextProcessor osztályt tartalmaz, amely a fájlok feldolgozásáért felelős.

### 4. Absztrakt osztályok

Az alapvető összehasonlítási logika egy absztrakt osztályban, a TextComparer osztályban található. Az alosztályok (például a SimpleTextComparer) implementálják az egyes szövegek összehasonlítását.

### 5. Eredmény kiírása

A program kiírja az összehasonlítás eredményét a felhasználónak. Az eredmény formátuma: "A fájlok [x]% -ban egyeznek."

# Osztályok és funkciók

## 1. TextComparer (Absztrakt osztály)

Az TextComparer osztály meghatározza az alapvető metódust, compare\_texts(self, text1, text2), amelyet az alosztályoknak kell implementálniuk.

- Paraméterek:
  - text1 (str): Az első szöveg.
  - text2 (str): A második szöveg.
- Visszatérési érték: A két szöveg hasonlósága százalékos formátumban (float).

## 2. TextProcessor

A TextProcessor osztály felelős a fájlok beolvasásáért és a szöveggé alakításáért.

- Funkciók:
  - load\_file(file\_path): Beolvassa a fájlt és visszaadja a szöveget.
  - prepare\_text(text): Előkészíti a szöveget az összehasonlításhoz (eltávolítja a felesleges szóközöket és figyelmen kívül hagyja a kis- és nagybetűket).
- Paraméterek:
  - file\_path (str): A fájl elérési útja.
- Visszatérési érték:
  - A fájl szövege (str).

## 3. SimpleTextComparer

Az SimpleTextComparer osztály végzi az egyszerű szöveges összehasonlítást.

- Funkciók:
  - compare\_texts(self, text1, text2): Összehasonlítja a két szöveget, és kiszámítja a Jaccard-hasonlóságot.
- Paraméterek:
  - text1 (str): Az első szöveg.
  - text2 (str): A második szöveg.
- Visszatérési érték:
  - A két szöveg hasonlósága százalékos formátumban (float).

#### 4. FileComparer

A FileComparer osztály a TextProcessor osztály segítségével végzi el a fájlok összehasonlítását.

- **Funkciók:**
  - `compare_files(file1, file2)`: Összehasonlítja két fájl tartalmát.
- **Paraméterek:**
  - `file1 (str)`: Az első fájl elérési útja.
  - `file2 (str)`: A második fájl elérési útja.
- **Visszatérési érték:**
  - A fájlok hasonlósága százalékos formátumban (float).

#### 5. ResultPrinter

A ResultPrinter osztály felelős az összehasonlítás eredményének kiírásáért.

- **Funkciók:**
  - `print_result(result)`: Kiírja az összehasonlítás eredményét.
- **Paraméterek:**
  - `result (float)`: Az összehasonlítás eredménye.
- **Visszatérési érték:**
  - Nincs visszatérési érték, az eredményt kiírja a képernyőre.

### Bónusz feladat

A program képes a fájlokat karakterenként is összehasonlítani, nemcsak szó szintjén, ezzel növelve a rugalmasságot.

### Műszaki részletek

A program a következő elveket alkalmazza:

- **SOLID**: A kód tervezése során a SOLID elveket követjük, biztosítva a moduláris és karbantartható megoldást.
- **KISS**: A kód egyszerű és könnyen érthető marad, elkerülve a túl bonyolult megoldásokat.
- **DRY**: Az ismétlődő kódot elkerüljük a megfelelő funkciók és osztályok létrehozásával.
- **YAGNI**: Csak a szükséges funkciókat implementáljuk, kerülve a túltervezést.