

# Programozási nyelvek Java

---

## 10. gyakorlat

### StringTokenizer

Egy String objektumot felbont mezőkre a megadott szeparátor elem mentén.

- A java.util csomagban található
- Konstruktorában meg kell adni egy String típusú szöveget és egy String típusú elválasztó szöveget
- **boolean hasMoreElements()**: megmondja, hogy található-e még elem a szövegben
- **boolean hasMoreTokens()**: megmondja, hogy található-e még elem a szövegben
- **Object nextElement()**: megadja a következő elemet
- **String nextToken()**: megadja a következő elemet
- **int countTokens()**: megadja, hogy összesen hány részre bontotta fel az inputot

### File

Ez az osztály reprezentál egy fájlt vagy egy könyvtárat.

- A java.io csomagban található
- Konstruktorában meg kell adni egy elérési utat
- **String getName()**: visszadja a fájl nevét
- **String getPath()**: visszaadja a fájl teljes elérési útját
- **boolean isFile()**: megadja, hogy a fájl-e
- **boolean isDirectory()**: megadja, hogy könyvtár-e
- **boolean exists()**: teszteli, hogy létezik-e
- **boolean canRead()**: megmondja, hogy olvasható-e
- **boolean canWrite()**: megmondja, hogy írható-e
- **boolean mkdir()**: létrehoz egy könyvtárat
- **boolean createNewFile()**: létrehoz egy fájlt
- **boolean delete()**: kitörli a fájlt
- **boolean deleteOnExit()**: kilépéskor törli a fájlt
- **boolean renameTo(File f)**: átnevezi a fájlt
- **boolean setReadOnly()**: csak olvashatóra állítja
- **String getParent()**: visszaadja a fájlrendszerbeli szülő nevét
- **File getParentFile()**: visszaadja a fájlrendszerbeli szülőt
- **String[] list()**: visszatér a fájlrendszerbeli tartalmával (csak nevekkkel)
- **File[] listFiles()**: visszatér a fájlokkal, amiket tartalmaz
- **File[] listFiles(FileFilter f)**: visszatér a fájlokkal, amiket tartalmaz és megfelel a szűrőnek

## FileFilter

A fájlok szűrését végzi. Például a FileFilter típusú objektum dönti el, hogy az aktuális könyvtárban lévő fájlok közül melyek jelenjenek meg a felhasználó számára egy JFileChooser esetén.

Az **accept** és a **getDescription** metódusait felül kell definiálni!

- javax.swing.filechooser csomagban található
- **boolean accept(File f)**: visszaadja, hogy a paraméterként kapott fájl megjelenjen-e a JFileChooser diagsablakban
- **String getDescription()**: megadja az elfogadott fájlok szöveges leírását

## Feladatok

1. Készíts egy olyan programot, ami az első paraméterben (argumentumban) megadott nevű fájlban található szövegben **(soronként újra)** megszámlolja, hogy hány „a” betűvel elválasztott rész található, és a második paraméterben megadott nevű fájlba beírja a részek számát (mindig új sorba). A fájl elejére még kerüljön be a fájl neve is.
2. Egy tömb elemei növekedő sorrendben beírt természetes számok. Keressünk meg ebben egy adott  $h$  értéket! (logaritmikus kereséssel). A keresés adja vissza az elem indexét, ha a tömb tartalmaz olyan elemet, különben adjon vissza -1-t.  
A tömb elemeit egy fájlban tároljuk el. A fájl első eleme az elemek száma, a többi pedig az elem (megadott elemszámnyi).
3. Egy adatszerkezet elemei egész számok. Rendezzük az elemeket növekvő sorrendbe!  
Az elemeket egy fájlban tároljuk el. A fájl első eleme az elemek száma, a többi pedig az elem.  
Az osztály neve Sort legyen.  
Legyen benne egy Read függvény, ami beolvassa a számokat és visszaadja az elemeket. Tartalmazzon még egy Sort függvény, ami rendezi az eredeti (beolvasott) elemeket, és elmenti egy új változóba (ne az eredetit írja felül!!!) (rendezett elemek).  
Tartalmazzon továbbá egy WriteConsol metódust, ami a kapott elemeket kiírja a következő formában: „3 | 5 | 45 | 97”  
Kell még egy Items és egy SortedItems getter metódusok, amik visszaadják az eredeti és a rendezett elemeket.  
Használd a honlapon található SortTester fájlt a teszteléshez!
4. Készíts egy programot, amely beolvassa a megadott könyvtárban levő összes „txt” kiterjesztésű fájlt és kiírja egy másik fájlba az ezekből kiválasztottak neveit.  
A program szóljon, ha nem könyvtárat vagy nem létező könyvtárat adtunk meg. (Sima üzenet a konzolra!)

Legyen a programnak egy menüje, ahol választhatunk a konzolos és a fájlból történő beolvasások közül.

Ha a konzolosat választjuk, akkor bekéri a könyvtár elérési útját. Ezután kilistázza sorszámmal a benne levő „txt” fájlokat (pl: 1 – alma.txt, ...). Ezután megjelenik egy újabb menü, amiben vagy megjelölhetünk egy fájlt (bekér egy sorszámot) vagy befejezzük. Az első választása esetén bekéri a megjelölni kívánt fájl sorszámát, majd ismét kiírja a menüt (esetleges újabb megjelölésért). A második menüpont esetében bekér egy fájlnevet, majd létrehozza „txt” kiterjesztéssel és beleírja a kiválasztott fájlok neveit.

Fájlbeolvasás esetén minden automatikusan történjen (azaz minden szükséges adat a fájlban megtalálható).

A fájl első sora az könyvtár elérési útja. Utána minden sorban egy sorszám (kiválasztottak sorszámai). Legvégén pedig a kiírandó fájl neve.