

## Feladatok fájlkezeléssel

1. Olvassuk be az `allatok.txt` szövegfájlt, és a tartalmát írassuk ki a képernyőre!
  - a) Először használjuk a `FileStream` és a `StreamReader` osztályokat!
  - b) Csak a `StreamReader` osztályt használjuk!
2. Olvassuk be az `allatok.txt` szövegfájlt, majd fűzzük hozzá a tigris sorát! A lezárás előtt ne felejtsük el üríteni a puffert! (Használjuk a `StreamReader` és a `StreamWriter` osztályokat. Az utóbbinál fontos lesz a `true` paraméter a hozzáfűzés miatt. Mi történik, ha ezt nem adjuk meg, illetve `false` értékű?
3. Olvassuk be a `proba.txt` fájlt, majd minden második sorát írassuk ki a képernyőre!
4. Generáltassunk 100 darab véletlenszámot 0 és 1000 között (a határokat is megengedve), majd írjuk ki azokat a `generaltszamok.txt` fájlba! Minden egyes szám új sorba kerüljön, és kapjon sorszámot, amelytől egy szóköz választja el. Ellenőrzésképpen nyissuk meg a fájlt, és a tartalmát írassuk ki a képernyőre!
5. Tudjuk, hogy a `szamok.txt` szövegfájl 3 soros, és minden sorban 5 egész szám van szóközzel elválasztva egymástól. A feladat egy új fájl (`forditva.txt`) létrehozása úgy, hogy minden sorban fordított sorrendben szerepeljenek a számok. Egyúttal a képernyőre is írassuk ki a fordított sorrendű számokat!
6. Írjunk programot, amely paraméterként megkapja egy szövegfájl nevét (és kiterjesztését) az elérési útvonalával együtt. Legyen benne kivételkezelés (beolvasás alatti hiba; nem található a fájl; nem található az alkönyvtár; saját kivétel dobása, ha nem volt megadva paraméter)!
  - a) Írassuk ki a fájl tartalmát zöld és fehér színnel, soronként váltogatva a kiírás színét!
  - b) A kiírás 20 soronként lapozva történjen!
7. Írjuk ki az egész számokat 0-tól 99-ig egy bináris fájlba, majd olvassuk vissza!