# 14. Fájlok 1.

## Bevezetés

A mai leckében egy szöveges fájlt szeretnénk beolvasni, és tartalmát kiíratni a képernyőre. Mivel azonban a fájlok használata során hibák történhetnek (pl. nem létező fájlt próbálunk megnyitni), először a hibakezeléssel kell megismerkednünk.

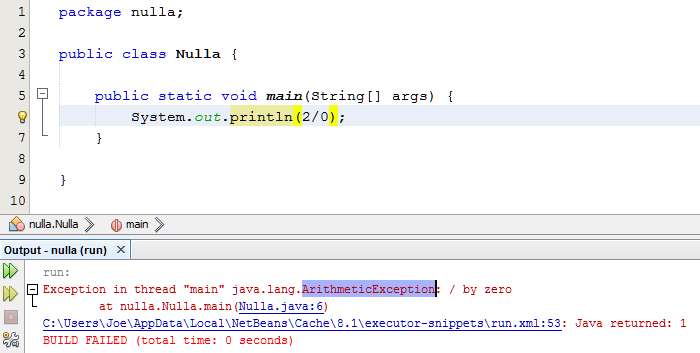
## Hibák a programban

Egy program lehetséges hibáit két csoportba sorolhatjuk:

* Szintaktikai hibák: ezeket piros hullámvonallal jelzi a NetBeans, és el sem indul a program, amíg ki nem javítjuk őket.
* Futási hibák: a program szintaktikailag helyes, elindul, de futás közben valamelyik művelet eredménye hiba lesz.

Futási hibák esetén **kivétel** (**exception**) történik a programban és a program hibaüzenettel leáll.

Nézzünk például egy nullával osztást: (Ezt a programot nem kell elkészítened!)



Minden kivétel egy osztály, amelynek van neve. Az ábra alsó részén kiemeltük ennek a nevét (ArithmeticException).

Kétféle kivétel van. Az egyik, amit nem kötelező kezelnünk, mint a fenti példában. Ilyen esetekben a program jó megírásával elkerülhető a hiba. Például nem osztunk nullával. (Unchecked exception)

A kivételek másik fajtája nem kerülhető el így. Például ilyen, amikor a felhasználó rossz fájlnevet ad meg, és megpróbáljuk megnyitni. Ezeket a kivételeket kötelező kezelni a programban. (Checked exception)

A kivételek kezelésére két lehetőségünk van:

* Vagy a metódus (amelyben a kivétel történhet) fejlécében, a paraméterek után jelezzük, hogy milyen kivétel fordulhat elő. Ekkor a metódus továbbadja a kivételt az őt hívó metódusnak (vagy a main esetén a Java virtuális gépnek).
* Vagy a metódusban kezeljük a kivételt. Ez úgy történik, hogy azokat az utasításokat, amelyek kivételt okozhatnak, egy try utasításba tesszük, és utána egy catch ágban megadjuk, hogy mi történjen hiba esetén. (Egy elágazásra hasonlít.)

## Szöveges fáj beolvasása

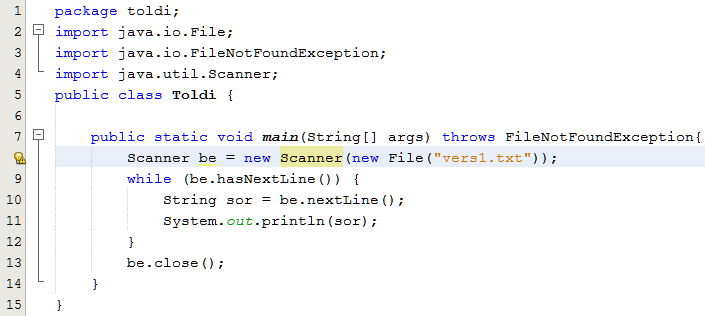
A *vers1.txt* fájl a Toldi első két versszakát tartalmazza UTF-8 kódolással, a *vers2.txt* pedig ugyanezt Windows 1250-es kódolással. A projekt létrehozása után, a program futtatása előtt ezeket a projekt mappájába kell másolni!

Ha a projekt mappájában van a fájl, akkor nem kell megadni útvonalat a fájl neve előtt. Ha máshol van, akkor meg kell adni, ilyen formában: "T:/vers1.txt"

Egy fájl megnyitásakor a FileNotFoundException kivétel következhet be, ha a program nem találja a fájlt. Ezt a kivételt kötelező kezelni.

Készítsünk egy olyan programot, amely megnyitja a *vers1.txt* fájlt, majd soronként beolvassa, és kiírja a képernyőre! A végén zárjuk be a fájlt!

A program első változata:



A main() metódus fejlécében (7. sor) jelezzük a kivételt, mert ez kötelezően kezelendő, és nem írtunk try utasítást. Enélkül nem lehet lefordítani a programot. Ráadásul importálni kell a megfelelő osztályokat (2-3.sor), de ehhez csak a Ctrl+Shift+I billentyűket kell megnyomnunk.

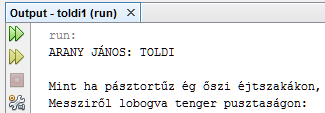
A szöveg beolvasásához egy be nevű Scanner típusú objektumot fogunk használni. A Scanner nem csak a billentyűzetről, hanem fájlból is tud olvasni. A létrehozásakor azonban (a System.in helyett) egy File típusú objektumot kell megadnunk a 8. sorban látható módon.

A Scanner objektum hasNextLine() metódusa azt adja meg, hogy van-e következő sor a fájlban. A sorokat addig olvassuk be és írjuk ki a képernyőre, amíg van következő sor.

A beolvasás után be kell zárnunk a fájlt a Scanner objektum close() metódusával.

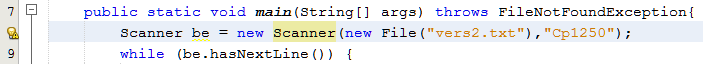
Egyébként egész vagy törtszámokat is ugyanúgy olvashatnánk be a fájlból, mint a billentyűzetről (nextInt(), nextDouble(), ...).

Másold a szöveges fájlokat a projekt mappájába, majd próbáld ki a programot!



## Kódolás megadása

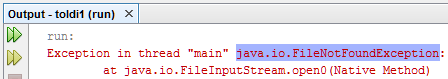
Írd át a programban a fájl nevét *vers2.txt*-re, és próbáld ki így is! Azért nem jelennek meg a sorok, mert a program a kódolás miatt nem látja őket. A megoldás a kódolás megadása a 8. sorban így:



Próbáld ki így is!

## Hibakezelés

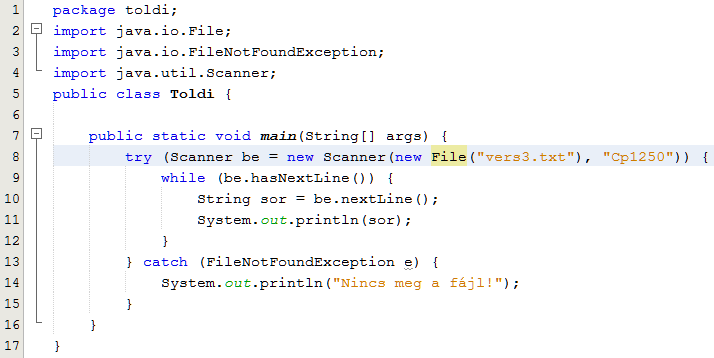
Írd át a programban a fájl nevét *vers3.txt*-re, és próbáld ki így is! Nézd meg a hibaüzenetet!



A *vers3.txt* fájl nem létezik, ezért a FileNotFoundException kivétel következik be, és leáll a program.



Nézzük meg, hogyan lehet kezelni a hibát:



Itt már nincs szükség a throws FileNotFoundException részre, mert a metódus kezeli a hibát.

A try után zárójelbe tesszük azokat az utasításokat, amelyek kivételt okozhatnak. Ha nem történik kivétel, akkor a try ág további részét is végrehajtja a program.

Ha kivétel történik, akkor kiválasztja a hozzá tartozó catch ágat, amelyben a megfelelő kivétel szerepel, és csak az abban szereplő utasításokat hajtja végre. (Most csak egy catch ág van, de lehetne több is.)

A végén a program automatikusan bezárja a zárójelben megnyitott fájlokat.

Próbáld ki így is, majd állítsd vissza fájlnevet *vers2.txt*-re!

## Feladat

Alakítsd úgy a programot, hogy a végén írja ki a sorok számát is! Minta:

