Kivételkezelés

Készítette: Vastag Atila

2017

Vegyük a következő esetet:

```
x: int = 10
y: int = 0

hanyados: float = x / y

print(hanyados)
```

hanyados: float = x / y
ZeroDivisionError: division by zero

- Nullával való osztás miatt kapjuk a hibaüzenetet

Példa II

Vegyük egy másik esetet:

```
input: str = "nem vagyok szam"
szam: float = int(input)

print(szam)
```

szam: float = int(input)
ValueError: invalid literal for int() with base 10: 'nem vagyok szam'

- A hibaüzenetet azért kapjuk mert a felhasználói bevitelnél nem egész számot adtunk meg!

Kivételek

Nyílván vannak olyan esetek, amikor az alkalmazásunk, bár gond nélkül lefordul, mégsem úgy fog működni, ahogy elképzeltük. Az ilyen „abnormális” működés kezelésére találták ki a kivételkezelést. Amikor az alkalmazásunk „rossz” állapotba kerül, akkor egy ún. kivételt fog dobni.

Ilyen problémák láthatóak a :

Példa I – nullával próbáltunk osztani, ami nem lehetséges

Példa II – ahol *int* típusú változóba próbáltunk *string* típusú változót eltárolni

Ilyennel már találkoztunk a listáknál is, amikor túlindexeltünk.

Természetesen mi azt szeretnénk, hogy valahogy kijavíthassuk ezt a hibát, ezért el fogjuk kapni a kivételt. Ehhez a művelethez három dologra van szükségünk: kijelölni azt a programrészt, ami dobhat kivételt, elkapni azt és végül kezeljük a hibát:

Kivétel keletkezésének két lehetősége van:

- Egyik lehetőség amikor a keretrendszer generálja hibát, amelyet ha nem kapunk el, akkor az operációs rendszertől kapjuk a hibaüzenetet és a program leáll
- Másik lehetőség, hogy mi is tudunk kivételt dobni, amennyiben valamilyen hibalehetőséget találunk programunk logikájában és ezt kezeljük

```
try:
        <utasítások >
( except < kivétel >, változó:
        <kivételkezel  utasítások >)
[ else
        <utasítások a korrekt programvégrehajtás esetére>]
```

```
try:
        f = open ( '/ tmp/ bar. txt ')
except IOError , ex:
        print(ex)
else :
        for line in f.readlines():
                print(line)
        f. close()
```

try:

< utasítások >

finally :

< műveletek >

Bármi is történik, a **try…final** konstrukció lehetővé teszi a helyzet kezelését még akkor is, ha egy kivétel (exception) generálódott.

A **finally** blokk kiértékelésére sor kerül egy **try** blokkbeli return előtt függetlenül attól, hogy generálódott-e kivétel vagy sem.

Figyelem : egy finally blokkbeli return kimaszkol egy try blokkbeli returnt.