

Python. Obsługa wyjątków

Wyjątek to sytuacja nienormalna, która pojawia się w trakcie wykonania programu, dlatego też w wielu językach programowania wprowadzono obsługę wyjątków. Pozwala ona zareagować na sytuacje, w których istnieje ryzyko niewykonania określonego zadania.

Do obsługi wyjątków głównie korzysta się z instrukcji try, except, else, finally, raise.

Przykład obsługi wyjątku:

```
x = 6
y = 0
try:
    z = x / y
except ZeroDivisionError:
    print("Nie dziel przez zero!")
```

Część programu narażonego na błąd należy objąć instrukcją **try**, która przechwyci wyjątek i pozwoli go obsłużyć. Po tej sekcji należy umieścić **except**, gdzie umieszczony będzie kod wykonywany w razie znalezienia błędu. Przy wykryciu błędu Python przeskoczy do części except gdzie wykona zawarte w nim instrukcje a następnie będzie działać dalej, nie wracając już jednak do sekcji try. Wszystkie instrukcje z sekcji try znajdujące się po wykrytym błędzie zostaną pominięte.

Drugi przykład:

```
try :
    plik = open ("testfile.txt" , "w")
    plik.write (" Test obsługi bledow ")
except IOError :
    print (" Bład : nie mozna odnalezc pliku lub odczytac danych ")
else :
    print ("Poprawnie dokonano zapisu do pliku ")
    plik.close()
```

Ten sam kod z wykorzystaniem finally:

try :

```
plik = open("testfile.txt", "w")
```

```
plik.write(" Test obsługi błędów ") try :
```

```
    plik.write (" Test obsługi błędów 2 ")
```

```
finally :
```

```
    print (" Zamykam plik ")
```

```
    plik.close()
```

```
except IOError :
```

```
    print (" Błąd : nie można odnaleźć pliku lub odczytać danych ")
```

W bloku finally umieszcza się instrukcje, które zostaną wykonane niezależnie od tego, czy w sekcji try zostanie wykryty błąd czy też nie, oraz nawet niezależnie od tego, czy wykonywanie kodu z bloku try zostanie przerwane przez `break` lub `return`. Komendę `finally` wykorzystuje się do wykonania koniecznych porządków. Z instrukcją `finally` nie można równocześnie stosować `else`. Instrukcja `try` musi posiadać co najmniej jeden blok `except` lub jeden blok `finally`, ale nigdy oba równocześnie.

Istnieje wiele "typów" wyjątków, oto kilka z nich:

Exception - Wszystkie wyjątki

SystemExit - Generowany przez `sys.exit()`

IOError - Błędy związane z operacjami na plikach lub operacjami I/O

EOFError - Zgłaszany po osiągnięciu końca pliku

KeyboardInterrupt - Generowany przez klawisze przerwania (zazwyczaj CTRL+C)

SyntaxError - Błąd składni

Więcej o wyjątkach :

<https://pl.python.org/docs/lib/module-exceptions.html>

<https://docs.python.org/3/library/exceptions.html>

Lista zadań:

1. Napisz program, który sprawdzi czy podany ciąg liczb przez użytkownika jest polskim kodem pocztowym. Jeśli nie jest program powinien zwrócić wyjątek. Każdy poprawnie wprowadzony kod pocztowy powinien zostać dodany na sam koniec do pliku o nazwie `kodypocztowe.txt`. Jak

zmieni się program, kiedy będziemy chcieli stworzyć program z wykorzystaniem amerykańskich kodów pocztowych.

2. Napisz program, który otwiera dwa pliki o nazwach podanych w programie. Jeśli argumentów nie podano, wówczas nazwy plików mają być pobrane od użytkownika. W dwóch plikach umieść osobne wiadomości prasowe znalezione na dowolnej stronie informacyjnej np. news z polityki i sportu. Program powinien wyświetlać wiersze z obu plików naprzemiennie, to znaczy: 1-szą linię z pierwszego pliku, 1-szą linię z drugiego pliku, 2-gą linię z pierwszego pliku, 2-gą linię z drugiego pliku, itd., aż do momentu, wyświetlenia ostatniego wiersza pliku zawierającego większą liczbę wierszy. Program powinien zapewniać obsługę błędów try... Wynik zapisz do pliku .txt