Zasięgi nazw

Zasięg nazw (scope of variables): „gdzie widać nazwę i do czego się ona odnosi".

Przestrzeń nazw (*namespace*): kolekcja nazw oraz obiektów, które są z nimi stowarzyszone. Służy do rozstrzygania, co dana nazwa oznacza w danej sytuacji.

Sytuacja 1. Każdy obiekt ma swoją przestrzeń nazw, obejmującą atrybuty danego obiektu:

```
z = 2 + 2j
z1 = z.real          # z - konkretny obiekt typu complex,
                     # w jego przestrzeni nazw jest
                     # stowarzyszenie "real" -> 2
                     # i stąd z.real to 2

z2 = z.conjugate()   # "conjugate" -> metoda (funkcja)
                     # i stąd z.conjugate - wywoływalny
```

Zasięgi nazw

Sytuacja 2. (pozornie nie związana z 1.) – ramki wywoływania.

- Ramka globalna – zawsze obecna, rozstrzyga znaczenie nazw spoza wywołania funkcji.
- Ramka lokalna (wywołania funkcji) – tworzona dla konkretnego wywołania funkcji.
- Nazwy utworzone poza wywołaniem funkcji należą do ramki globalnej.
- Nazwy utworzone w wywołanej funkcji (w tym parametry formalne) należą do ramki tego wywołania.
- Nazwy w różnych ramkach mogą się powtarzać, lecz nazywać inne obiekty.

Odnosząc się do nazw: najpierw sprawdzana jest ramka aktualnego wywołania funkcji (jeśli istnieje), następnie ramka globalna.

Zasięgi nazw

Sytuacja 3. (pozornie nie związana z 2.) – przestrzenie nazw modułów.

```
import math # tworzy obiekt typu module pod nazwą math
            # z własną przestrzenią nazw
            # np. "sin" -> funkcja

# przykład w PyCharmie
import my_module # tworzy moduł nazwany my_module
                 # w ramce globalnej
                 # wykonuje my_module.py
                 # utworzone nazwy trafiają do
                 # przestrzeni nazw my_module
# warianty
from my_module import a
from my_module import f as func
```

Ale w istocie: „ramka globalna” = przestrzeń nazw (aktualnego) modułu! Slogan: kolejność wyszukiwania nazw: LEGB.