Objekty …pokračování

Funkce definovaná v třídě může být

1) funkce instance/objektu
2) statická funkce
3) funkce třídy

1) Funkce patřící objektu

- patří objektu (volá se prostřednictvím objektu)
- má přístup k jeho funkcím a proměnným

= to, co jsme viděli doposud

2) Statická metoda/funkce

- patří třídě a ne objektu (volá se pomocí třídy i objektu)
- nemá přístup k proměnným objektu
  (žádný objekt ani nemusí existovat)

```
class Trida:

    @staticmethod
    def sedm():
        return 7

print( Trida.sedm() ) # neexistuje žádný objekt
t = Trida()
print( t.sedm() )
```

```
7
7
```

3) Metoda třídy (class method)

- patří třídě a ne objektu (volá se prostřednictvím třídy)
- nemá přístup k proměnným objektu
  (žádný objekt ani nemusí existovat)
- má přístup k proměnným třídy

```
class AA:
    pocet = 0
    @classmethod
    def MetodaTridy(cls): # parametr cls
        cls.pocet += 1
        print( cls.pocet, end=" " )

AA.MetodaTridy()
AA.MetodaTridy()
```

```
1 2
```

Proměnná v třídě může být

1) proměnná patřící instanci
2) proměnná patřící třídě

1) proměnná patřící instanci

Zatím všechno, co jsme viděli (až na předchozí příklad),
i když ve skutečnosti…

2) proměnná patřící třídě

```
class KLAS:
    pocet = 0
    def __init__(self):
        KLAS.pocet += 1
        self.ID = KLAS.pocet
    def tisk(self):
        print( self.ID, end=" " )

a = KLAS()
a.tisk()
b = KLAS()
b.tisk()
```

```
1 2
```

…ale není to tak jednoduché

```
class KLAS:
    pocet = 0
    def __init__(self):
        KLAS.pocet += 1
    def f(self):
        #self.pocet += 1
        return self.pocet

t = KLAS()
print( t.f(), t.f(), t.f() )
s = KLAS()
print( s.f(), s.f(), s.f() )
```

a)
```
#self.pocet += 1:

1 1 1
2 2 2
```

b)
```
self.pocet += 1:

2 3 4
3 4 5
```

Pokud v tom nemáte jasno, tak to nepoužívejte!

Virtuální metody

V odvozené třídě lze předefinovat metodu tak,
že všechna její volání (i ze starších tříd a metod)
budou volat tu novou.

V Pythonu jsou všechny metody virtuální.

Podrobnosti – u nějakého jiného jazyka.

Někdy jindy, případně vůbec

objektový návrh…

dekorátory

class-factory

Příklady…