

Objektovo
orientované
programovanie
- *pokročilí*

Učiteľ:
Ing. Jozef Wagner PhD.

Učebnica:
<https://oop.wagjo.com/>

OPGP

Pokročilí 9

1. Triedy a metódy
2. Inštančné metódy a atribúty
3. Triedne metódy a atribúty
4. Statické metódy

Trieda - class

Konštruktor triedy má názov `__init__`

Prvý argument je referencia na vytváraný objekt `self`

Novú inštanciu (nový objekt) vytvoríme volaním triedy ako metódu

```
class Obdlznik:
```

```
    def __init__(self, x, y):
```

```
        self.x = x
```

```
        self.y = y
```

```
maly_obdlznik = Obdlznik(2, 3)
```

Inštančné metódy a atribúty

Atribúty nemusíme deklarovať vopred.

Metódy majú prvý argument **self**

Novú inštanciu (nový objekt) vytvoríme volaním triedy ako metódu

```
class Obdlznik:
```

```
    def obvod(self):
```

```
        return 2 * (self.x + self.y)
```

```
print(f"Obvod obdlznika: {maly_obdlznik.obvod()}")
```

Dunder inštančná metóda `__str__`

Špeciálne metódy upravujú správanie triedy.

`__str__` slúži na user-friendly (užívateľsky prívetivý) výpis objektu vo forme reťazca.

```
class Obdlznik:
```

```
    def __str__(self):
```

```
        return f"{self.x}x{self.y}"
```

```
print(f,,Obdlznik {maly_obdlznik}")
```

Triedne atribúty a metódy

Začínajú anotáciou `@classmethod` a prvý argument majú `cls`.

```
class Teplomer:  
    jednotka = "C"
```

```
    def __init__(self, teplota):  
        self.teplota = teplota
```

```
    @classmethod  
    def zmen_jednotku(cls, nova):  
        cls.jednotka = nova
```

```
moj_teplomer = Teplomer(37)  
print(Teplomer.jednotka)  
print(moj_teplomer.jednotka)
```

Statické metódy

Funkcie definované v triede, ktoré nemajú žiaden prístup ani k objektu ani ku triede.

Pri ich definícii sa používa anotácia `@staticmethod`

```
class Matematika:  
    @staticmethod  
    def sucet(a, b):  
        return a + b
```

```
    @staticmethod  
    def mocnina(x):  
        return x * x
```