



C2184
Úvod do programování
v Pythonu

Lekce 9

Úvod do OOP

Úvod do OOP, třídy a objekty

C2184 Úvod do programování v Pythonu
podzim 2016

Třída a objekt

Stanislav Geidl
Národní centrum pro výzkum biomolekul
Masarykova univerzita



- **třída (class)** - definice objektu, předepisuje metody a atributy objektu
- **objekt (object)** - instance třídy
- **metoda (method)** - funkce objektu
- **atribut (attribute)** - proměnná objektu
- **vlastnost (property)** - vlastnost objektu (funkce get a set)
- **self** - odkaz na samu sebe
- **statické třídy a funkce** - nevytváříme instance
- **privátní atributy a metody** - v Pythonu označovány tím, že začínají `__`, tyto proměnné a funkce by neměli být použity mimo definici třídy



```
""" třída Tuzka """  
class Tuzka:  
    __tuha = 100.0  
    tvrdost = 1  
    def pis(self):  
        print ("Pišu")  
        __tuha -= .5  
  
""" instance mojeTuzka """  
mojeTuzka = Tuzka()  
mojeTuzka.pis()  
  
mojeTuzka.tvrdost = 2  
print (mojeTuzka.tvrdost)
```

Vlastnosti

```
class Tuzka:
    __tuha = 100.0
    __guma = False
    @property
    def guma(self):
        return __guma
    @guma.setter
    def guma(self, novahodnota):
        __guma = bool(novahodnota)
    @property
    def tuha(self):
        return __tuha
```

```
mojeTuzka = Tuzka()
mojeTuzka.guma = 2
print(mojeTuzka.guma)
print(mojeTuzka.tuha)
""" toto nefunguje:
mojeTuzka.guma = 2
"""
```





- **__init__** - konstruktor
- **__del__** - destruktork
- **__str__** - textový přepis třídy/objektu
- **__doc__** - krátká dokumentace třídy/objektu
- **__len__** - funkce pro výpočet délky
- tyto funkce se automaticky volají při vytvoření nebo zničení instance, při volání funkcí `len()` nebo `str()/print()`



```
class Tuzka:
    __tuha = 100.0
    tvrdost = 1
    def __init__(self):
        print("Já_jsem_konstruktor")
    def __len__(self):
        return __tuha
    def pis(self):
        print("Pišu")
        __tuha -= .5

mojeTuzka = Tuzka()
mojeTuzka.pis()
print(len(mojeTuzka))
```



- abstrakce
- zapozdření
- skládání
- delegování
- dědičnost
- polymorfismus



```
class Savec:
    def __init__(self):
        self.srst = True
        self.vyska = 100
    def pozdrav(self):
        print(self._pozdrav)

class Clovek(Savec):
    def __init__(self):
        self.srst = False
        self._pozdrav = "Ahoj"
```

```
ja = Clovek()
ja.pozdrav()
```

```
""" toto nebude fungovat
print(ja.vyska)
"""
```



```
class Savec:
    def __init__(self):
        self.srst = True
        self.vyska = 100
    def pozdrav(self):
        print(self._pozdrav)

class Clovek(Savec):
    def __init__(self):
        self.srst = False
        self._pozdrav = "Ahoj"

class Pes(Savec):
    def __init__(self):
        super().__init__()
        self._pozdrav = "Baf_baf"

ja = Clovek()
ja.pozdrav()
Baryk = Pes()
Baryk.pozdrav()
```

