

Třídy a objekty

```
class Zviratko:
    def __init__(self, jmeno):
        self.jmeno = jmeno

    def snez(self, jidlo):
        print(self.jmeno, 'jí', jidlo)
```

Vlastní třída, neboli typ
Inicializace objektu, může přebírat argumenty
Nastavení atributu objektu

```
class Kotatko(Zviratko):
    def zamnoukej(self):
        print(self.jmeno, 'mňau!')
```

Dědičnost (kotátko je druh zvířátka)
Metoda specifická pro kotátka

```
class Prasatko(Zviratko):
    def zakvic(self):
        print(self.jmeno, 'kví!')

    def snez(self, jidlo):
        self.zakvic()
        super().snez(jidlo)
        print(self.jmeno,
              'se po jídle válí v bahně!')
```

Dědičnost (prasátko je druh zvířátka)
Metoda specifická pro prasátka



Přepisování metody předka
Volání vlastní metody
Volání metody jako na předkovi

```
prasatko = Prasatko('Ťulda')
```

Vytvoření objektu (instance) třídy

```
prasatko.snez('žalud')
```

Volání metody

```
karlicka = Prasatko('Karlička')
print(karlicka.jmeno)
karlicka.jmeno = 'Karla'
```

Objektů jedné třídy může být víc
Můžeme se dívat na jejich atributy
A taky je měnit

```
mikes = Kotatko('Mikeš')
```

Polymorfismus

```
for zviratko in [prasatko, karlicka, mikes, Kotatko('Mourek')]:
    zviratko.snez('papání')
```

Nevíme jestli to je prasátko nebo kotátko,
ale to nevádí, tohle funguje

```
isinstance(zviratko, Zviratko)
type(zviratko) == Zviratko
```

Je instancí zvířátka? Ano
Je třídou objektu Zviratko? Ne

Typová kontrola