

Objektovo  
orientované  
programovanie  
- *pokročilí*

Učiteľ:  
**Ing. Jozef Wagner PhD.**

Učebnica:  
<https://oop.wagjo.com/>

# OPGP

## Pokročilí 11

1. Zapuzdrenie
2. Modifikátory prístupu
3. Property
4. Hra adventúra

# Zapuzdrenie

Objekt v sebe spája stav (atribúty) a správanie (metódy) do jedného celku

Zapuzdrenie sa používa na skrytie interných detailov objektu

## Modifikátory prístupu

Základné zapuzdrenie je riešené pomocou obmedzenia prístupu k atribútom a metódam triedy

Interné (protected) a súkromné (private) atribúty a metódy majú podčiarníky v názve

# Getter, Setter a Deleter

Je zvykom atribúty nezverejňovať, ale mať ich súkromné (private)

Na prístup k súkromným atribútom sa vytvárajú getter a setter metódy

Vieme ich naimplementovať pomocou obyčajných inštančných metód

# Getter, Setter a Deleter

```
class Osoba:
    def __init__(self, meno):
        self._meno = meno

    def get_meno(self):
        return self._meno

    def set_meno(self, meno):
        if not meno:
            raise ValueError("Meno chýba")
        self._meno = meno
```

```
fero = Osoba("Fero")

print(fero.get_meno())

fero.set_meno("Jozo")

print(fero.get_meno())
```

# Property

Elegantnejší spôsob na getter a setter metódy

Akoby priamy prístup k atribútu, ale v skutočnosti sa budú volať getter a setter metódy

- getter - čítanie hodnoty atribútu
- setter - zápis hodnoty do atribútu
- deleter - zmazanie atribútu pomocou del

```
atribut = property(fget, fset, fdel, doc)
```

# Property

```
class Osoba:
    def __init__(self, meno):
        self._meno = meno

    def get_meno(self):
        return self._meno

    def set_meno(self, meno):
        if not meno:
            raise ValueError("Meno chýba")
        self._meno = meno

    meno = property(get_meno, set_meno)
```

```
fero = Osoba("Fero")

print(fero.meno)

fero.meno = "Jozo"

print(fero.meno)
```

# @property decorator

Syntax sugar

Elegantnejší spôsob na getter a setter metódy

@property, @atribut.setter, @atribut.deleter

**V kóde musí byť ako prvá uvedená getter metóda, pred setter a deleter metódami**



# @property decorator

```
class Osoba:  
    def __init__(self, meno):  
        self._meno = meno
```

```
    @property  
    def meno(self):  
        return self._meno
```

```
    @meno.setter  
    def meno(self, meno):  
        if not meno:  
            raise ValueError("Meno chyba")  
        self._meno = meno
```

```
fero = Osoba("Fero")
```

```
print(fero.meno)
```

```
fero.meno = "Jozo"
```

```
print(fero.meno)
```

# Hra adventúra

Hra v konzole, hráč sa pohybuje medzi miestnosťami

Hra čaká na náš príkaz. Príkazy sa zadávajú vo forme `sloveso podstatné_meno`, napr. `go vychod`, čo je príkaz na to, aby sme sa presunuli do miestnosti, ktorá je východným smerom.

Vytvoríme si balík `adventura` a 4 moduly s triedami:

- Hráč - `player.py`
- Miestnosť - `room.py`
- Stav hry - `hra.py`
- Logika hry - `engine.py`
- Vstupný bod do hry bude v súbore `__main.py__`