**Dědičnost, method overriding, function overloading**

**Dědičnost**

* Způsob ustanovení vztahu mezi třídy nebo objekty
* Dědičnost napomáhá k znovu-použitelnosti kódu
* Simple vysvětlení: Jev, ve kterém jedna třída přebírá vlastnosti a funkce od jiné třídy a zároveň ji rozšiřuje o další funkce a vlastnosti, který zároveň specifikují původní třídu

PŘ:

Třída ZVÍŘE

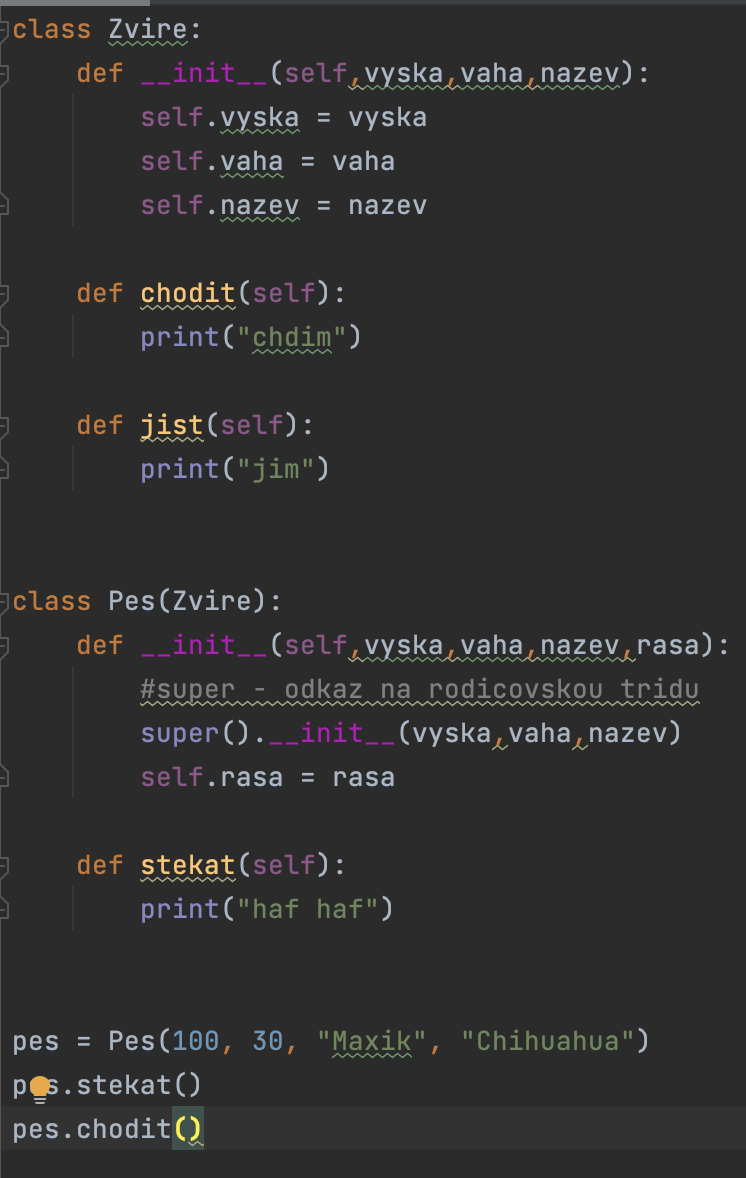
Třída PES

Třída PES bude přebírat vlastnosti od třídy ZVÍŘE (váha, výška, název, FUNKCE Chodit, FUNKCE Jíst)

A děděním, třída pes bude obsahovat identické vlastnosti a funkce.

Dále se ve třídě PES například navíc implementuje vlastnost PLEMENO a funkce Štěkat.

V pythonu „super“ na rodičovskou třídu

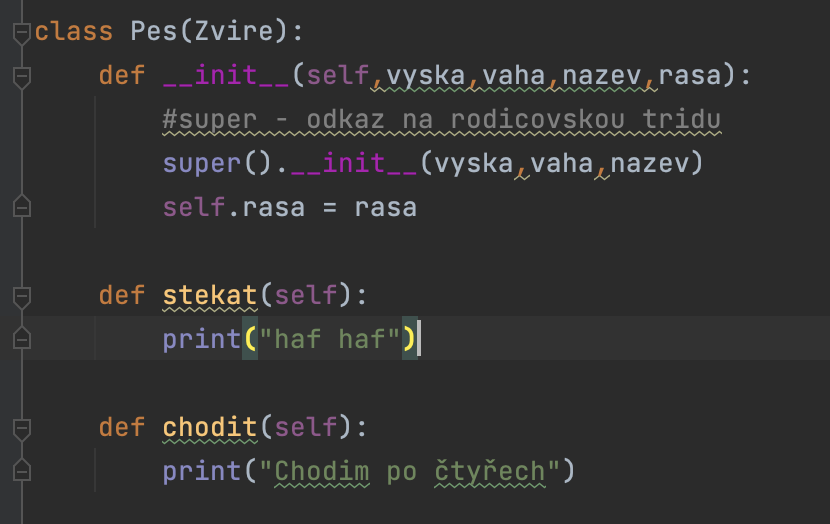


**Method overriding**

* Změna funkcionality metody, která byla původně zděděná od rodičovské třídy

PŘ:

Metodu z třídy ZVÍŘE Chodit přepíšeme v třídě PES, aby měla stejný název a stejný počet vstupních atributů



**Function overloading**

* Možnost mít více funkcí se stejným jménem za podmínky jiného počtu vstupních atributů

PŘ:

Ve třídě PES, metodu Štěkat napíšeme s jinýma vstupníma atributama, první – původní z předešlého příkladu, jako další – štěkání s 1 vstupním atributem kde se bude určovat s jakým přízvukem štěká, a jako další – počet štěkání a prizvuk zaroven

!!! V PYTHONU ZADNY FUCNCTION OVERLOADING NENI !!!

Jako priklad uvadim C# a vypocty

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated